DESCRIPTION

INFORMATION PROCESSING SYSTEM AND METHOD

Technical Field

The present invention relates to an information processing system, an information processing method, an information recording medium, and a program distributing medium, and particularly, to a system and a method for distributing an encryption processing key in a system involving an encryption processing. Particularly, the invention relates to an information processing system, an information processing method, an information recording medium, and a program distributing medium, which uses a tree-structured hierarchical key distributing system, reconstructs a hierarchical key distributing tree according to a distributing device to reduce data quantity contained in a distributing key block to thereby reduce a distributing message quantity, relieves loads of a content key distribution or data distribution when various keys are renewed, and can hold safety of data.

Background Art

Recently, various software data (which will be hereinafter called contents) such as game programs, voice data, image data, and so on have been actively circulated through a network such as an internet, or storage media capable of being circulated such as DVD, CD, etc. These circulation contents are reproduced by

reception of data by a PC (Personal Computer) owned by a user or game apparatus, or by mounting a memory medium, or are stored in a recording device within a recording and reproducing apparatus attached to PC and the like, for example, a memory card, a hard disk and the like, the contents being utilized by new reproducing from the stored medium.

Information apparatuses such as a video game apparatus, PC and the like have an interface for receiving the circulation contents from a network or for getting access to DVD, CD and the like, and further have control means necessary for reproducing the contents, and RAM, ROM and the like used as a memory region for programs and data.

Various contents such as music data, image data, or programs are called from a memory medium by user's instructions from the information apparatus such as a game apparatus, PC and the like used as a reproducing apparatus or user's instructions through input means connected, and are reproduced though information apparatus or a display, a speaker and the like connected.

Many software contents such as game programs, music data, image data and the like are generally held in their distribution rights by owners and sales agents. Accordingly, in distribution of these contents, there is a predetermined using limitation, that is, the use of software is granted to only proper users so that reproduction without permission is not made. That is, generally, the constitution taking security into consideration is employed.

One procedure for realizing the limit of use to users is an encryption processing of distributed contents. Namely, for example, various contents such as voice data, image data, game programs and the like encrypted through an internet or the like are distributed, and means for decrypting the encrypted contents distributed, that is, a decryption key is given to only persons confirmed to be a proper user.

Encrypted data can be returned to decrypted data that can be used by decrypting processing in accordance with the predetermined procedure. Data encrypting using a decryption key for decrypting processing, and a decrypting method, using an encrypted key for encryption processing of information as described have been heretofore well known.

There are a variety of kinds of forms of data encrypting and decrypting method using an encryption key and a decryption key, but there is, as one example therefor, a system called a so-called common key encryption system. In the common key encryption system, with an encryption key used for encrypting processing for data and a decryption key used for decrypting data made to be common, a common key used for these encrypting processing and decrypting is given to a proper user so as to eliminate the data access by an invalid user. As a typical system of the system as described, there is DES (Data Encryption Standard).

The encryption key and the decryption key used for the encrypting processing and decrypting as described above can be obtained by applying a

unidirectional function such as a hash function on the basis of a pass-word or the like, for example. The unidirectional function herein termed is a function which is very difficult to obtain an input conversely from an output. For example, the unidirectional function is applied with a pass-word determined by a user as an input, and the encryption key and the decryption key are produced on the basis of the output. It is substantially impossible, from the encryption key and the decryption key thus obtained, to conversely obtain a pass-word which is an original datum thereof.

A system making processing by an encryption key used for encryption and processing by a decryption key used for decrypting different algorithm is a system so-called a public key encryption system. The public key encryption system is a method using a public key that can be used by an unspecific user, in which with respect to an encrypted document for a specific individual, encrypting processing is carried out using a public key issued by the specific individual. The document encrypted by the public key can be subjected to decrypting processing merely by a private key corresponding to the public key used for the encrypting processing. The private key is owned merely by the individual who issued the public key, and the document encrypted by the public key can be decrypted merely by the individual having the private key. A typical public key encryption system is a RSA (Rivest-Shamir-Adleman) encryption. By making use of such an encryption system, there can be provided a system for enabling decrypting encrypted contents merely

for a proper user.

In the content distributing system as described above employs many constitutions in which contents are encrypted and stored in the recording media such as a network, or DVD, CD and the like to provide them for users, and to provide a content key for decrypting encrypted contents for only a proper user. There is proposed a constitution in which a content key for preventing invalid copies of the content key itself is encrypted to provide it to a proper user, and an encrypted content key is decrypted using a decryption key owned by only the proper user to enable using the content key.

The judgment whether or not a user is proper is generally carried out by executing authenticating processing before distribution of contents or content keys, for example, between a content provider who is a transmitter of contents and a user's device. In general authenticating processing, confirmation is made of a mating party, and a session key effective only for communication is produced. When authentication is established, data, for example, contents or a content key is encrypted using the produced session key for communication. The authenticating system includes mutual authentication using a common key encryption system, and an authentication system using a public key system. In the authentication using a common key, a common key in the system wide is necessary, which is inconvenient at the time of renewal processing. Further, in the public key system, computation load is large and necessary memory quantity increases, and the provision of such a

processing means on each device is not a desirable constitution.

Disclosure of the Invention

It is an object of the present invention to provide an information processing system, an information processing method, an information recording medium, and a program distributing medium, which enables transmission of data safely to a proper user without relying on mutual authentication processing between a transmitter and a receiver of data as described above, and reconstructs a hierarchical key distribution tree according to a distribution tree to reduce data quantity contained in a distribution key block to thereby reduce data quantity of an encryption key, reduces load of data transmission, and enables reduction of processing for obtaining an encryption key in each device.

An information processing system according to the present invention is one for distributing encrypted message data capable of being used only in not less than one device selected, the individual device comprising: encryption processing means for holding a different key set of a node key peculiar to each node in a hierarchical tree structure having a plurality of different devices as leaves and a leaf key peculiar to each device and executing decrypting process of the encrypted message data distributed to a device using the key set; wherein the encrypted message data distributed to the device has data constitution to be encrypted with a renewal node key obtained in a decrypting process of an enabling key block (EKB) including

encrypted key data into which the renewal node key into which at least one of the node keys in a group constituted by nodes and leaves connected at subordinate of a top node which is one node of the hierarchical tree structure is encrypted by the node key or the leaf key in the group, and the enabling key block (EKB) includes a data part constituted by the encrypted key data and a tag part as position discrimination data of the encrypted key data in the hierarchical tree structure.

Further, in one embodiment of the information processing system according to the present invention, the encrypted key data included in the enabling key block (EKB) is data into which a node key constituting the hierarchical tree structure is encrypted using a subordinate node key or a subordinate leaf key, and position discrimination data stored in the tag part is constituted as a tag indicating whether there is the encrypted key data at subordinate left and right node or leaf position of a node position of each of not less than one encrypted key data stored in the enabling key block (EKB) or not.

Further, in one embodiment of the information processing system according to the present invention, the encrypted key data included in the enabling key block (EKB) is constituted on the basis of only keys corresponding to a node or a leaf of a reconstructed hierarchical tree reconstructed by selecting paths constituting a simplified 2-branched type tree with terminal nodes or leaves with which the enabling key block (EKB) can be decrypted at the lowest stage to omit unnecessary nodes, and position discrimination data stored in the tag part includes data

indicating whether the encrypted key corresponding to the tag of the enabling key block (EKB) is stored or not.

Further, in one embodiment of the information processing system according to the present invention, the encrypted key data included in the enabling key block (EKB) is constituted on the basis of only a key corresponding to a node or a leaf of a reconstructed hierarchical tree reconstructed by selecting paths constituting a simplified 2-branched type tree with terminal nodes or leaves with which the enabling key block (EKB) can be decrypted at the lowest stage to omit unnecessary nodes, and position discrimination data stored in the tag part includes tags indicating whether encrypted key data at left and right node or leaf position at subordinate of a node position of each of not less than one encrypted key data stored in the enabling key block (EKB), and data indicating whether the encrypted key corresponding to the tag is stored or not.

Further, in one embodiment of the information processing system according to the present invention, the reconstructed hierarchical tree is a tree constituted by selecting a sub-root which is a top node of an entity defined as a subset tree of devices having a common element.

Further, in one embodiment of the information processing system according to the present invention, the encrypted key data included in the enabling key block (EKB) is constituted, in a simplified multi-branched type tree having terminal node or leaf with which the enabling key block (EKB) can be decrypted at the lowermost

stage, on the basis of only keys corresponding to a top node and terminal nodes or leaves of a reconstructed hierarchical tree reconstructed by selecting paths directly connecting the terminal nodes or leaves and a top of the multi-branched type tree to omit an unnecessary node, and position discrimination data stored in the tag part includes data indicating whether an encrypted key corresponding to the tag of the enabling key block (EKB) is stored or not.

Further, in one embodiment of the information processing system according to the present invention, the reconstructed hierarchical tree is a tree having not less than three branches connecting the top node constituting the simplified multi-branched type tree with terminal nodes or leaves constituting the simplified tree directly.

Further, in one embodiment of the information processing system according to the present invention, the encryption processing means in the device has a constitution for sequentially extracting the encrypted key data with data of the tag part in the enabling key block (EKB), executing decrypting process to obtain the renewal node key, and executing decryption of the encrypted message data with the renewal node key obtained.

Further, in one embodiment of the information processing system according to the present invention, the message data is a content key that can be used as a decryption key for decrypting content data.

Further, in one embodiment of the information processing system according

to the present invention, the message data is an authentication key used in the authentication process.

Further, in one embodiment of the information processing system according to the present invention, the message data is a key for generating an integrity check value (ICV) of the content.

Further, in one embodiment of the information processing system according to the present invention, the message data is a program code.

Further, an information processing method according to the present invention is one for distributing encrypted message data capable of being used only in not less than one selected devices, comprising: an enabling key block (EKB) generating step of generating an enabling key block (EKB) including a data part including encrypted key data into which the renewal node key into which at least one of the node keys in a group constituted by nodes and leaves connected at subordinate of a top node which is one node of the hierarchical tree structure is renewed is encrypted with a node key or a leaf key in the group, and a tag part which is position discrimination data in the hierarchical tree structure of encrypted key data stored in the data part; and a message data distribution step for generating message data encrypted with the renewal node key to distribute it to a device.

Further, one embodiment of the information processing method according to the present invention comprises a decrypting processing step of executing decrypting process to the encrypted message data using the key set in a device

holding a different key set of a node key peculiar to each node in the hierarchical structure and a leaf key peculiar to each device.

Further, in one embodiment of the information processing method according to the present invention, the enabling key block (EKB) generating step includes a step of encrypting a node key constituting the hierarchical tree structure using a subordinate node key or a subordinate leaf key to generate the encrypted key data, and a step of generating a tag indicating whether there is encrypted key data at a node or leaf position at subordinate left and right positions of a node position of each of not less than one encrypted key data stored in the enabling key block (EKB) or not to store it in the tag part.

Further, in one embodiment of the information processing method according to the present invention, the enabling key block (EKB) generating step includes a step of generating a reconstructed hierarchical tree by selecting paths constituting a simplified 2-branched type tree with a terminal node or leaf capable of decrypting the enabling key block (EKB) at the lowest stage to omit unnecessary nodes; a step of generating an enabling key bock (EKB) on the basis of only a key corresponding to a constitution node or leaf of the reconstructed hierarchical tree; and a step of storing data indicating whether an encrypted key corresponding to a tag of the enabling key block (EKB) is stored in the tag part or not.

Further, in one embodiment of the information processing method according to the present invention, the step of generating the reconstructed hierarchical tree is

tree generating processing executed by selecting a sub-root which is a top node of entity defined as a subset tree of devices having a common element.

Further, in one embodiment of the information processing method according to the present invention, the enabling key block (EKB) generating step includes a step of generating, in the simplified branched type tree with a terminal node or leaf capable of decrypting the enabling key bock (EKB) at the lowest stage, the reconstructed hierarchical tree reconstructed by selecting a path for directly connecting the terminal node or leaf with the top of the multi-branched type tree; and a step of storing data indicating whether an encrypted key corresponding to a tag of the enabling key bock (EKB) is stored in the tag part or not.

Further, in one embodiment of the information processing method according to the present invention, the reconstructed hierarchical tree generated in the step of generating the reconstructed hierarchical tree is generated as a tree having not less than three branches having a top node constituting a simplified multi-branched type tree and a terminal node or leaf constituting a simplified tree connected directly.

Further, in one embodiment of the information processing method according to the present invention, the decrypting processing step includes a renewal node key obtaining step of obtaining the renewal node key by sequentially extracting encrypted key data stored in the data part on the basis of position discrimination data stored in the tag part of the enabling key block (EKB) to sequentially execute decrypting process; and a message data decrypting step for executing decryption of

the encrypted message data with the renewal node key.

Further, in one embodiment of the information processing method according to the present invention, the message data is a content key capable of being used as a decryption key for decrypting the content data.

Further, in one embodiment of the information processing method according to the present invention, the message data is an authentication key used in the authentication process.

Further, in one embodiment of the information processing method according to the present invention, the message data is a key of generating an integrity check value (ICV) of contents.

Further, in one embodiment of the information processing method according to the present invention, the message data is a program code.

Further, an information recording medium according to the present invention is one having data stored. The recording medium stores an enabling key block (EKB) including a data part including encrypted key data into which the renewal node key into which at least one of the node keys in a group constituted by nodes and leaves connected under a top node which is one node of the hierarchical tree structure is renewed is encrypted with a node key or a leaf key in the group, and a tag part which is position discrimination data in the hierarchical tree structure of encrypted key data stored in the data part, and message data encrypted by the renewal node key.

Further, in one embodiment of the information recording medium according to the present invention, the encrypted key data included in the enabling key block (EKB) is data into which the node key constituting the hierarchical tree structure is encrypted using a subordinate node key or a subordinate leaf key; and the position discrimination data stored in the tag part is constituted as a tag indicating whether there is key data at the node of leaf position at the subordinate left and right positions of the node position of each of not less one encrypted key data stored in the enabling key block (EKB).

Further, in one embodiment of the information recording medium according to the present invention, the encrypted key data included in the enabling key block (EKB) is constituted on the basis of only a key corresponding to a node or a leaf of a reconstructed hierarchical tree reconstructed by selecting paths constituting a simplified 2-branched type tree with a terminal node or leaf capable of decrypting the enabling key block (EKB) at the lowest stage to omit unnecessary nodes; and the position discrimination data stored in the tag part includes data indicating whether an encrypted key corresponding to the tag of the enabling key block (EKB) is stored or not.

A program distributing medium according to the present invention is one for distributing a computer program to execute on a computer system a process of generating an enabling key block (EKB) into which a renewal node key into which at least one of the node keys in a group constituted by nodes and a leaves connected

under the top node which is one node of the hierarchical tree structure is renewed is encrypted with a node key or a leaf key in the group. The computer program includes a step of generating a reconstructed hierarchical tree by selecting a path constituting a simplified 2-branched type tree with a terminal node or a leaf capable of decrypting the enabling key block (EKB) at the lowest stage to omit an unnecessary node; a step of generating the enabling key block (EKB) on the basis of only a key corresponding to a constitution node or leaf of the reconstructed hierarchical tree; and a step of storing data indicating whether an encrypted key corresponding to a tag of the enabling key block (EKB) is stored or not.

In the constitution of the present invention, the encryption key distributing constitution of the hierarchical structure of the tree structure is used to suppress the distributing message quantity necessary for key renewal as small as possible. That is, the key distribution method in which each apparatuses is arranged in each leaf by n-division is used whereby for example, a content key which is an encryption key of content data or an authentication key used in authentication processing or a program code are distributed along with an enabling key block through recording medium or a communication circuit.

Further, the enabling key block is constituted by an encrypted key data part and a tag part showing a position of the encrypted key, whereby data quantity is reduced to enable rapid execution of decrypting processing in a device. According to the present constitution, only the proper device is able to distribute decodable

data safely.

It is noted that the program distributing medium according to the present invention is a medium for distributing a computer program in the form that can be read by a computer to a general computer system capable of executing, for example, various program codes. The medium includes recording media such as CD, FD, MO, etc., or a transfer medium such as a network, whose form is not particularly limited.

Such a program distributing medium defines a cooperative relationship in terms of constitution or function between a computer program and a distributing medium in order to realize a function of a predetermined computer program in a computer system. In other words, a computer program is installed in a computer system through the distributing medium to exhibit the cooperative operation in the computer system to obtain the operation and effect similar to another aspects.

The other objects, features and advantages of the present invention will be apparent from the detailed description with reference to the embodiments and the accompanying drawings of the present invention.

Brief Description of the Drawings

FIG. 1 is a view for explaining an example of constitution of an information processing system according to the present invention.

FIG. 2 is a block diagram showing an example of constitution of a recording

and reproducing apparatus that can be applied in the information processing system according to the present invention.

FIG. 3 is a tree constitution view for explaining encryption processing of various keys and data in the information processing system according to the present invention. FIGS. 4A and 4B are views each showing an example of an enabling key block (EKB) used in distribution of various keys and data in the information processing system according to the present invention.

FIG. 5 is a view showing an example of distribution and an example of decrypting processing using an enabling key block (EKB) of content keys in the information processing system according to the present invention.

FIG. 6 is a view showing an example of a format of an enabling key block (EKB) in the information processing system according to the present invention.

FIGS. 7A to 7C are views each for explaining a constitution of a tag of an enabling key block (EKB) in the information processing system according to the present invention.

FIGS. 8A and 8B are views each showing an enabling key block (EKB) and an example of data constitution for distributing content keys and contents in the information processing system according to the present invention.

FIG. 9 is a view showing an example of processing in a device in case of distributing an enabling key block (EKB), content keys, and contents in the information processing system according to the present invention.

FIG. 10 is a view for explaining the situation how to cope with the case where an enabling key block (EKB) and contents are stored in the information processing system according to the present invention.

FIGS. 11A and 11B are views each showing comparison between processing for sending an enabling key block (EKB) and contents in the information processing system according to the present invention and a conventional sending processing.

FIG. 12 is a view showing an authentication processing sequence according to an applicable common key encryption system in the information processing system according to the present invention.

FIG. 13 is a view (1) showing an enabling key block (EKB), a data constitution for distributing an authentication key, and a processing example by a device in the information processing system according to the present invention.

FIG. 14 is a view (2) showing an enabling key block (EKB), a data constitution for distributing an authentication key, and a processing example by a device in the information processing system according to the present invention.

FIG. 15 is a view showing an authentication processing sequence by a public key encryption system applicable in the information processing system according to the present invention.

FIG. 16 is a view showing a processing for distributing an enabling key block (EKB) and content keys using the authentication principle by a public key

encryption system in the present invention.

FIG. 17 is a view showing a processing for distributing an enabling key block (EKB) and encrypted program data in the information processing system according to the present invention.

FIG. 18 is a view showing an example of MAC value production used in production of a content integrity check value (ICV) applicable in the present invention.

FIG. 19 is a view (1) showing a data constitution for distributing an enabling key block (EKB) and an ICV producing key, and an example of a processing in a device in the information processing system according to the present invention.

FIG. 20 is a view (2) showing a data constitution for distributing an enabling key block (EKB) and an ICV producing key, and an example of a processing in a device in the information processing system according to the present invention.

FIGS. 21A and 21B are views each for explaining a copy preventive function where an applicable content integrity check value (ICV) is stored in a medium in the present invention.

FIG. 22 is a view for explaining a constitution for controlling an applicable content integrity check value (ICV) separately from a content storage medium in the present invention.

FIG. 23 is a view for explaining an example of category classification of a hierarchical tree structure in the information processing system of the present

invention.

FIGS. 24A and 24B are views each for explaining a producing process of a simplified enabling key block (EKB) in the information processing system of the present invention.

FIGS. 25A and 25B are views each for explaining a producing process of an enabling key block (EKB) in the information processing system of the present invention.

FIGS. 26A and 26B are views each for explaining a simplified enabling key block (EKB) (Example 1) in the information processing system of the present invention.

FIGS. 27A and 27B are views each for explaining a simplified enabling key block (EKB) (Example 2) in the information processing system of the present invention.

FIGS. 28A to 28C are views each for explaining an entity control constitution of a hierarchical tree structure in the information processing system of the present invention.

FIGS. 29A to 29C are views each for explaining, in detail, an entity control constitution of a hierarchical tree structure in the information processing system of the present invention.

FIGS. 30A and 30B are views each for explaining an entity control constitution of a hierarchical tree structure in the information processing system of

the present invention.

FIG. 31 is a view for explaining a reserve node in an entity control constitution of a hierarchical tree structure in the information processing system of the present invention.

FIG. 32 is a view for explaining a new entity registration sequence in an entity control constitution of a hierarchical tree structure in the information processing system of the present invention.

FIG. 33 is a view for explaining a relationship between a new entity and a host entity in an entity control constitution of a hierarchical tree structure in the information processing system of the present invention.

FIGS. 34A and 34B are views each for explaining a sub-EKB used in an entity control constitution of a hierarchical tree structure in the information processing system of the present invention.

FIGS. 35A to 35D are views each for explaining a device revoke processing in an entity control constitution of a hierarchical tree structure in the information processing system of the present invention.

FIG. 36 is a view for explaining a device revoke processing sequence in an entity control constitution of a hierarchical tree structure in the information processing system of the present invention.

FIGS. 37A and 37B are views each for explaining a renewal sub-EKB at the time of device revoke in an entity control constitution of a hierarchical tree

structure in the information processing system of the present invention.

FIGS. 38A to 38D are views each for explaining an entity revoke processing in an entity control constitution of a hierarchical tree structure in the information processing system of the present invention.

FIG. 39 is a view for explaining an entity revoke processing sequence in an entity control constitution of a hierarchical tree structure in the information processing system of the present invention.

FIG. 40 is a view for explaining a relationship between a revoke entity and a host entity in an entity control constitution of a hierarchical tree structure in the information processing system of the present invention.

FIG. 41 is a view for explaining a capability setting in an entity control constitution of a hierarchical tree structure in the information processing system of the present invention.

FIG. 42 is a view for explaining a capability setting in an entity control constitution of a hierarchical tree structure in the information processing system of the present invention.

FIGS. 43A and 43B are views each for explaining a capability control table for controlling a key issuing center (KDC) in the information processing system of the present invention.

FIG. 44 is an EKB producing processing flowchart on the basis of a capability control table for controlling a key issuing center (KDC) in the

information processing system of the present invention.

FIG. 45 is a view for explaining a capability notice processing at the time of new entity registration in the information processing system of the present invention.

Best mode for Carrying out the Invention

[Outline of System]

FIG. 1 shows an example of a content distributing system to which the data processing system of the present invention can be applied. The content distributing side 10 transmits a content or a content key encrypted to various content reproducible apparatuses on the content receiving side 20. The apparatus on the content receiving side 20 decrypts an encrypted content or a content key received to obtain a content or a content key, and carries out reproduction of image data and voice data or execution of various programs. The exchange of data between the content distributing side 10 and the content receiving side 20 is executed through a network such as an internet or through a circulatable recording medium such as DVD, CD.

The data distributing means on the content distributing side 10 includes an internet 11, a satellite broadcasting 12, a telephone circuit 13, media 14 such as DVD, CD, etc., and on the other hand, the devices on the content receiving side 20 include a personal computer (PC) portable apparatuses 23 such as a portable device

(PD), a portable telephone, PDA (Personal Digital Assistants), etc., a recording and reproducing unit 24 such as DVD, CD players, and a reproduction exclusive-use unit 25 such as a game terminal. In these devices on the content receiving side 20, contents distributed from the content distributing side 10 are obtained from communication means such as a network, or from a media 30.

[Constitution of Device]

FIG. 2 shows a block diagram of a recording and reproducing device 100 as one example of devices on the content receiving side 20 shown in FIG. 1. The recording and reproducing device 100 has an input/output I/F (Interface) 120, a MPEG (Moving Picture Experts Group) codec 130, an I/F (Interface) 140 provided with A/D, D/A converter 141, an encryption processing means 150, ROM (Read Only Memory) 160, CPU (Central Processing Unit) 170, a memory 180, and a drive 190 for a recording medium 195, which are connected to each other by a bus 110.

The input/output I/F 120 receives a digital signal constituting various contents such as an image, voice, a program, etc. supplied from the outside to output it to the bus 110, and receives a digital signal of the bus 110 to output it to the outside. The MPEG codec 130 decrypts MPEG coded data supplied through the bus 110 to output it to the input/output I/F 140, and MPEG-decrypts a digital signal supplied from the input/output I/F 140 to output it to the bus 110. The input/output I/F 140 contains an A/D, D/A converter 141 therein. The input/output I/F 140 receives an analog signal as a content supplied from the outside, which is subjected

to A/D (Analog Digital) conversion by the A/D, D/A converter 141 whereby the signal is output as a digital signal to the MPEG codec 130, and a digital signal from the MPEG codec 130 is subjected to D/A (Digital Analog) conversion by the A/D, D/A converter 141, which is output as an analog signal to the outside.

The encryption processing means 150 is constituted form, for example, one chip LSI (Large Scale Integrated circuit), to execute encrypting, decrypting processing or authentication processing of a digital signal as a content supplied through the bus 110, and output encrypted data and decrypted data to the bus 110. The encryption processing means 150 can be also realized by not only the one chip LSI but by a combination of various soft wares or hard wares. The constitution of the processing means formed from the software configuration will be described later.

ROM 160 stores program data processed by the recording and reproducing device. The CPU 170 executes programs stored in the ROM 160 and the memory 180 to thereby control the MPEG codec 130 and the encryption processing means 150. The memory 180 is for example, a non-volatile memory, which stores a program that is executed by the CPU 170, data necessary for operation of CPU 170, and a key set used in the encryption processing executed by the device. The key set will be explained later. The drive 190 drives the recoding medium 195 capable of recording and reproducing digital data to thereby read (reproduce) digital data from the recording medium 195 to output it to the bus 110, and supplies digital data

supplied through the bus 110 to the recording medium 195 for recording.

The recording medium 195 is a medium capable of storing digital data, for example, an optical disk such as DVD, CD, an optical magnetic disk, a magnetic disk, a magnetic tape, or a semiconductor memory such as RAM, and in the present embodiment, the medium can be detachably mounted on the drive 190. However, the recording medium 195 may be housed in the recording and reproducing device 100.

The encryption processing means 150 shown in FIG. 2 may be constituted as a single one-chip LSI, and may employ a constitution that is realized by a combination of a software and a hardware.

[Tree structure as a key distributing constitution]

Next, the constitution for holding an encryption processing key in each device and a data distributing constitution where encrypted data are distributed from the content distributing side 10 shown in FIG.1 to each device on the content receiving side 20 will be described using FIG. 3.

Numbers 0 to 15 shown in the lowest stage in FIG. 3 are individual devices on the content receiving side 20. That is, each leaf of the hierarchical tree structure shown in FIG. 3 corresponds to a device.

Each of devices 0 to 15 stores a key set comprising a key assigned to a node from own leaf to a root (a node key) and a leaf key of each leaf, in the hierarchical tree shown in FIG. 3, at the time of manufacture or at the time of shipment, or

afterwards. K0000 to K1111 shown in the lowest stage of FIG. 3 are respectively leaf keys assigned to devices 0 to 15, and keys from KR to K111 described in the second node from the lowest stage are node keys.

In the constitution shown in FIG. 3, for example, a device 0 has a leaf key K0000 and node keys K000, K00, K0, KR. A device 5 has K0101, K010, K01, K0, KR. A device 15 has K1111, K111, K11, K1, KR. In the tree of FIG. 3, only 16 devices 0 to 15 are described, and the tree structure is shown as a systematic constitution to left and right well balanced of a 4-stage constitution. However, much more devices may be constituted in the tree, and the parts of the tree may have the different number of stages.

Further, each device included in the tree structure shown in FIG. 3 includes various recording media, for example, DVD, CD, MD of the embedded type or the type detachably mounted on the device, or devices of various types using a flash memory or the like. Further, various application service may coexist. In addition to the coexisting constitution of various devices and various application, the hierarchical tree structure which is a content or a key distributing constitution shown in FIG. 3 is applied.

In the system in which various devices and applications coexist, for example, a portion surrounded by the dotted line in FIG. 3, that is, the devices 0, 1, 2 and 3 are set as a single group using the same recording medium. For example, with respect to the device included in the group surrounded by the dotted line,

processing is executed such that a common content is encrypted and sent from a provider, a content key used in common to devices is sent, or payment data for content charges is also encrypted and output from each device to a provider or a settlement organization. The organization for carrying out data transmit-receiving to and from the devices such as a content provider or a settlement organization executes processing for sending the portion surrounded by the dotted line of FIG. 3, that is, data collectively with the device 0, 1, 2, 3 as one group. A plurality of such groups are present in the tree of FIG. 3. The organization for carrying out data transmit-receiving to and from devices such as a content provider or a settlement organization functions as message data distributing means.

Node keys and leaf keys may be controlled collectively by a single key control center, or may be controlled every group by message data distributing means such as a provider, or a settlement organization for carrying out transmit-receiving of various data with respect to groups. These node keys and leaf keys are subjected to renewal processing when a key is leaked. This renewal processing is executed by a key control center, a provider or a settlement organization.

In this tree structure, as will be apparent from FIG. 3, three devices 0, 1,2, 3 included in one group hold common keys K00, K0, KR as a node key. By utilizing this node key common constitution, for example, a common content key can be distributed to only devices 0, 1, 2, 3. For example, if the node key K00 itself held in

common is set as a content key, only the devices 0, 1, 2, 3can be set as a common content key without executing new sending of key. Further, a value Enc(K00, Kcon) obtained by encrypting a new content key Kcon by a node key K00 is distributed to the devices 0, 1, 2, 3 through a network or by being stored in the recording medium, only the devices 0, 1, 2, 3 can decryption the encrypted Enc(K00, Kcon) using a common node key K00 held in the respective devices to obtain a content key: Kcon. The Enc(Ka, Kb) indicates data into which Kb is encrypted by Ka.

Further, where at the time t, keys : K0011, K001, K00, K0, KR owned by the device 3 are analyzed by a hacker and then exposed, it is necessary for protecting data transmit-received in a system (a group of devices 0, 1, 2, 3) to separate the device 3 from the system. To this end, node keys: K001, K00, K0, KR are respectively renewed to new keys K(t)001, K(t)00, K(t)0, K(t)R, which renewed keys to be notified to the devices 0, 1, 2. Here, K(t)aaa indicates a renewal key of Kaaa of generation : t.

The distributing processing of renewal key will be described. Renewal of key is executed by storing a table constituted by block data called an enabling key block (EKB: Enabling Key Block) shown in FIG. 4A in a network, for example, or in a recording medium to supply them to the devices 0, 1, 2. The enabling key block (EKB) is constituted by a decryption key for distributing a key newly renewed to a device corresponding to each leaf constituting a tree structure as shown in FIG. 3.

The enabling key block (EKB) is sometimes called a key renewal block (KRB: Key Renewal Block).

In the enabling key block (EKB) shown in FIG. 4A, only the device in which a node key need to be renewed is constituted as block data having a data constitution that can be renewed. An example of FIGS. 4A and 4B shows, in the devices 0, 1 and 2 in the tree structure shown in FIG. 3, block data formed for the purpose of distributing a renewal node key of generation t. As will be apparent from FIG. 3, the device 0 and the device 1 require K(t)00, K(t)0, K(t)R as renewal node keys, and the device 2 requires K(t)001, K(t)00, K(t)0, K(t)R as renewal node keys.

As shown in EKB of FIG. 4A, a plurality of encrypted keys are included in EKB. The encrypted key in the lowest stage is Enc(K0010, K(t)001). This is a renewal node key K(t)001 encrypted by a leaf key K0010 of the device 2, and the device 2 is able to decrypt this encrypted key by its leaf key to obtain K(t)001. By using K(t)001 obtained by decrypting, an encrypted key Enc(K(t)001, K(t)00) in the second stage from bottom can be decrypted to obtain a renewal node key K(t)00. Sequentially, an encrypted key Enc(K(t)00, K(t)0) in the second stage from top of FIG. 4A is decrypted to obtain a renewal node key K(t)0, and an encrypted key Enc(K(t)0, K(t)R) in the first stage from top of FIG. 4A is decrypted to obtain K(t)R. On the other hand, in the device K 0000, K0001, a node key K000 is not included to be renewed, and a key necessary for a renewal node key is K(t)00, K(t)0, K(t)R. The device K0000.K0001 decrypts an encrypted key Enc(K000,

K(t)00) in the third stage from top of FIG. 4A to obtain K(t)00, and thereafter, an encrypted key Enc(K(t)00, K(t)0) in the second stage from top of FIG. 4A is decrypted, and an encrypted key Enc(K(t)0, K(t)R) in the first stage from top of FIG. 4A is decrypted to obtain K(t)R. By doing so, the devices 0, 1, 2 can obtain a renewed key K(t)R. The index in FIG. 4A shows the absolute address of a node key and a leaf key used as a decryption key.

Where renewal of a node key : K(t)0, K(t)R in the upper stage in the tree structure shown in FIG. 3 is unnecessary, and a renewal processing of only the node key K00 is necessary, an enabling key block (EKB) in FIG. 4B can be used to distribute a renewal nod key K(t)00 to the devices 0, 1, 2.

EKB shown in FIG. 4B can be used, for example, to distribute a new content key in common in a specific group. Concretely, it is supposed that the devices 0, 1, 2, 3 shown by the dotted line in FIG. 3 use a recording medium, and a new common content key K(t)con is necessary. At this time, Enc(K(t)00, K(t)con) into which new common content key: K(t)con is encrypted with K(t)00 into which a common node key K00 of the devices 0, 1, 2 is renewed is distributed with EKB shown in FIG. 4B. By this distribution, distribution of data not decrypted in the apparatus of other groups such as a device 4 becomes enabled.

That is, if the devices 0, 1, 2 decrypt the encrypted sentence using K(t)00 obtained by processing EKB, a content key at the time t K(t)con can be obtained.

[Distribution of a content key using EKB]

FIG. 5 shows, as an example of processing for obtaining a content key at the time t K(t)con, a processing of a device 0 which receives, through a recording medium, data Enc(K(t)00, K(t)con into which a new common content key K(t)con is encrypted using K(t)00 and EKB shown in FIG. 4B. That is, this is an example in which encrypted message data by EKB is a content key K(t)con.

As shown in FIG. 5, a device 0 uses generation : EKB at generation: t stored in the recording medium and a node key K000 stored in advance by itself to produce a node key K(t)00 by the EKB processing similar to that described above. Further, a renewal content key K(t)con is decrypted using a renewal node key K(t)00 decrypted, and is encrypted by a leaf key K0000 owned by itself and stored in order to use it later.

[Format of EKB]

FIG. 6 shows an example of format of the enabling key block (EKB). A version 601 is a discriminator showing the version of the enabling key block (EKB). The version has a function for showing a corresponding relation between a function for discriminating latest EKB and a content. The depth shows the number of hierarchies of a hierarchical tree with respect to a device of the distributing destination of the enabling key block (EKB). A data pointer 603 is a pointer for indicating a position of data part in the enabling key block (EKB), and a tag pointer 604 is a pointer for indicating a position of a tag part, and a signature pointer 605 is a pointer for indicating a position of signature.

A data part 606 stores, for example, data having a node key to be renewed encrypted. For example, it stores various encrypted keys in connection with a renewal node key as shown in FIG. 5.

A tag part 607 is a tag for indicating a positional relationship of encrypted node keys and leaf keys stored in the data part. An attaching rule of this tag will be described with reference to FIGS. 7A to 7C. FIGS. 7A to 7C show an example for sending the enabling key block (EKB) described previously in FIG. 4A as data. The data at that time is as shown in FIG. 7B. An address of a top node included in an encrypted key at that time is used as a top node address. In this case, since a renewal key of a root key K(t)R is included, a top node address is KR. At this time, for example, data Enc(K(t)0, K(t)R) in the uppermost stage is at a position shown in a hierarchical tree shown in FIG. 7A. Next data is Enc(K(t)00, K(t)0), which is at a position under on the left hand of the previous data in the tree. Where data is exist, a tag is set to 0, and where data is not exist, a tag is set to 1. The tag is set as (left (L) tag, right (R) tag). Since data is exist at left of data at the top stage Enc(K(t)0, K(t)R), L tag = 0, and since data is not exist to right, R tag = 1. Tags are set to all the data to constitute a row of data and a row of tags shown in FIG. 7C.

The tag is set in order to show at which position of the tree structure data Enc(Kxxx, Kyyy) is positioned. Since the key data Enc(Kxxx, Kyyy) ... are mere enumerated data of simply encrypted keys, a position on the tree of an encrypted key stored as data can be discriminated by the aforementioned tag. For example,

data constitution as in the following can be provided using the node index placed in correspondence to the encrypted data like the constitution described in FIGS. 4A and 4B previously without using the aforementioned tag:

0: Enc(K(t)0, K(t)root)

00: Enc(K(t)00, K(t)0)

000: Enc(K(t)000, K(t)00)

...

However, the constitution using such an index as described results in lengthy data to increase data quantities, which is not preferable in the distribution through a network. On the other hand, the aforementioned tag is used as index data showing a key position whereby a key position can be discriminated with less data quantity.

Returning to FIG. 6, the EKB format will be further described. The signature is an electronic signature executed, for example, by a key control center, a content provider, a settlement organization or the like which issued the enabling key block (EKB). The device which received EKB confirms by authentication of signature that it is an enabling key block (EKB) issued by a valid enabling key block (EKB) issuer,

[Content Key Using EKB and Distribution of Contents]

While in the aforementioned example, a description was made of an example in which only the content key is sent along with EKB, a description will be made hereinafter of the constitution in which a content encrypted by a content key, and a

content key encrypted by a content encrypted key along with a content key encryption key encrypted by EKB are sent.

FIGS. 8A and 8B show this data constitution. In the constitution shown in FIG. 8A, Enc(Kcon, content) 801 is data in which a content is encrypted by a content key(Kcon), Enc(KEK, Kcon) 802 is data in which a content key (Kcon) is encrypted by a content key-encryption key (KEK: Key Encryption key), and Enc(EKB, KEK) 803 is data in which a content key-encryption key KEK is encrypted by an enabling key block (EKB).

Here, the content key-encryption key KEK may be a node key (K000, K00 ...) or a root key (KR) itself, and may be a key encrypted by a node key (K000, K00 ...) or a root key (KR).

FIG. 8B shows an example of constitution where a plurality of contents are recorded in media, which makes use of the same Enc(EKB, KEX) 805. In such a constitution as described, the same Enc(EKB, KEK) is not added to each data, but data showing a linking destination linked to Enc(EKB, KEK) is added to each data.

FIG. 9 shows an example of a case where a content encryption key KEK is constituted as a renewal node key K(t)00 obtained by renewed the node key K00 shown in FIG. 3. In this case, if in a group surrounded by the dotted frame in FIG. 3, the device 3 is revoked, for example, due to the leak of a key, data having an enabling key bock (EKB) shown in FIG. 9 and data into which a content key (Kcon) is encrypted by a content key encryption key (KEK = K(t)00), and data into

which a content is encrypted by a content key (Kcon) are distributed to members of the other groups, that is, devices 0, 1, 2 whereby the devices 0, 1, 2 can obtain the content.

The right side in FIG. 9 shows the decrypting procedure in the device 0. The device 0, first, obtains a content key encryption key (KEK = K(t)00) by decrypting process using a leaf key K000 held by itself from the received enabling key bock. Then, the device 0 obtains a content key Kcon decrypted by the K(t)00, and further carries out decrypting by the content key Kcon. The device 0 can use the content as a result of the above process. The devices 1, 2 are also able to obtain a content key encryption key (KEK=K(t)00) by processing EKB by the different procedures and are able to use the content similarly.

The devices 4, 5, 6 ... of the other groups shown in FIG. 3 are not able to obtain a content key encryption key (KEK = K(t)00) using a leaf key and a node key held by themselves even if they receive the same data (EKB) as mentioned above. The device 3 revoked is likewise not able to obtain the content key encryption key (KEK = K(t)00) by a leaf key and a node key, and only the device having the proper right is able to decrypt and use the content .

If the distribution of a content key making use of EKB is used, in a manner as described, the encrypted content which only valid right holder can decrypt can be distributed safely.

An enabling key block (EKB), a content key, an encrypted content or the like

has a constitution capable of providing distribution safely through a network, but the enabling key block (EKB), the content key and the encrypted content can be also stored in a recording medium such as DVD, CD and provided to a user. In this case, if constitution is made such that a content key obtained by decrypting an enabling key block (EKB) stored in one and the same recording medium is used for decrypting the encrypted content stored in the recording medium, distribution process of an encrypted content that can be used only with a leaf key and a node key held in advance by the valid right holder only, that is, content distribution for which a usable user's device is limited can be realized by a simple constitution.

FIG. 10 shows an example of constitution in which an enabling key block (EKB) is stored together with an encrypted content are stored in a recording medium. In the example shown in FIG. 10, stored in the recording medium are contents C1 to C4, data with the enabling key block corresponding to each stored content placed in correspondence thereto, and an enabling key block of version M (EKB - M). For example, EKB - 1 is used to produce a content key Kcon1 having a content C1 encrypted , and for example, EKB - 2 is used to produce a content key Kcon2 having a content C2 encrypted. In this example, an enabling key bock of version M (EKB - M) is stored in a recording medium. Since contents C3, C4 is placed in correspondence to the enabling key block (EKB - M), contents of the contents C3, C4 can be obtained by decrypting the enabling key block (EKB - M). Since EKB - 1, EKB - 2 are not stored in a disk, it is necessary to obtain EKB - 1,

EKB - 2 necessary for decrypts the respective content keys by new distribution means, for example, network distribution or distribution by a recording medium.

FIGS. 11A and 11B show a comparative example between a content key distribution by using EKB and conventional content key distribution where a content key is circulated among a plurality of devices. FIG. 11A shows the conventional constitution, and FIG. 11B shows an example making use of an enabling key block (EKB) according to the present invention. In FIGS. 11A and 11B, Ka (Kb) indicates data in which Kb is encrypted by Ka.

As shown in FIG. 11A, processing has been heretofore carried out in which validity of a data transmit-receiver is confirmed, authentication processing and authentication and key exchange (AKE) are executed between devices to co-own a session key Kses used in encrypting process of data transmission, and a content key Kcon is encrypted by the session key Kses under the condition that the authentication is established to effect transmission.

For example, in PC shown in FIG. 11A, it is possible to decrypt a content key Kses encrypted by a session key received by the session key to obtain Kcon, and further possible to encrypt Kcon obtained by a stored key Kstr held by PC itself to store it in own memory.

In FIG. 11A, processing is necessary in which even where data is desired to be distributed in the form capable of being used for only a recording device 1101 shown in FIG. 11A, when PC or a reproducing device is present, authentication

process as shown in FIG. 11A is executed so that content keys are encrypted by the respective session keys to effect distribution. The PC or the reproducing device is likewise able to use a session key produced in the authentication process and co-owned to decrypt an encrypted content key and obtain a content key.

On the other hand, in an example making use of an enabling key block (EKB) shown in the lower stage of FIG. 11B, an enabling key block (EKB), and data (Kroot (Kcon)) having a content key Kcon encrypted by a node key or a root key obtained by processing the enabling key block (EKB) are distributed from a content provider, whereby the content key Kcon can be decrypted and obtained by only the apparatus capable of processing EKB distributed.

Accordingly, for example, the useable enabling key block (EKB) is produced only on the right end in FIG. 11B, and the enabling key block (EKB), and data having a content key Kcon encrypted by a node key or a root key obtained by EKB processing are sent together whereby the PC, the reproducing apparatus or the like present cannot execute processing of EKB by a leaf key or node key owned by itself. Accordingly, the useable content key can be distributed to only the valid device safely without executing processes such as authentication process between the data transmit-receive devices, the production of a session key, and the process for encrypting a content key Kcon by the session key.

Where the useable content key is desired to be distributed to PC, a recording and reproducing unit also, an enabling key block (EKB) capable of being processed

is produced and distributed to thereby obtain a common content key.

[Distribution Of Authentication Key Using Enabling Key Block (EKB) (Common Key System)]

In the distribution of data used in the enabling key block (EKB) or a key described above, since an enabling key block (EKB) and a content or a content key which are transferred between devices always maintain the same encryption form, there is the possibility that an invalid copy is produced due to the so-called replay attack, which steals and records a data transmission channel and transfer it later again. For preventing such an attack as described, there is effective means for executing authentication process and key exchange process similar to those of prior art between data transfer devices. Now, a description is made of the constitution in which an authentication key Kake used when the authentication process and key exchange process are executed is distributed to a device using the aforementioned enabling key block (EKB) whereby the authentication process in conformity with a common key system having a common authentication key as a safe private key is executed. That is, this is an example in which encrypted message data by EKB is used as an authentication key.

FIG. 12 shows a mutual authentication method (IS0/IEC 9798-2) using a common key encryption system. While in FIG. 12, DES is used as the common key encryption system, other systems may be used as long as they are the common key encryption system . In FIG. 12, first, B produces the random number Rb of 64 bits,

and Rb and ID (b), which is own ID, are transmitted to A. A which receives them newly produces the random number Ra of 64 bits, and data are encrypted using a key Kab in the CBC mode of DES in order to Ra, Rb and Rc to transmit them to B. The key Kab is a key to be stored in a recording element as a private key common to A and B. According to the encrypting processing by the key Kab using the CBC mode of DES, for example, in the processing using DES, an initial value and Ra are subjected to exclusive OR; in the DES encryption part, the key Kab is used for encrypting to generate an encrypted text E1and continuously, the encrypted text E1 and Rb are subjected to exclusive OR; in the DES encryption part, a key Kab is used for encrypting, and encrypted text E2 and ID (b) are subjected to exclusive OR; and in the DES encryption part, a key Kab is used for encrypting to generate transmission data (Token-AB) by an encrypted text E3 produced.

B, which received the above data, decrypts the received data by a key Kab (authentication key) likewise stored in a recording element as a common private key. A decrypting method of received data, first, decrypts an encrypted text E1 by an authentication key Kab to obtain the random number Ra. Next, an encrypted text E2 is decrypted by an authentication key Kab, and the result therefrom and E1 are subjected to exclusive OR to obtain Rb. Finally, an encrypted text E3 is decrypted by an authentication key Kab, and the result therefrom and E2 are subjected to exclusive OR to obtain ID (b). Authentication is made if Ra and ID (b) out of Ra, Rb and ID (b) thus obtained are coincided with one transmitted by B. When passed

this authentication, B authenticates that A is valid.

Next, B produces a session key (Kses) to be used after authentication (Producing method: To use the random number). Then, Rb, Ra, Kses are encrypted in that order using an authentication key Kab in the CBC mode of DES and are returned to A.

A, which received the above data, decrypts the received data by an authentication key Kab. A decrypting method of the received data is similar to the decrypting process of B, which is therefore omitted in its detail. Authentication is made if Rb and Ra out of Rb, Ra and Kses thus obtained are coincided with one transmitted by A. When passed the authentication, A authenticates that B is valid. After authentication of mating parties each other, the session key Kses is used as a common key for secrete communication after authentication.

Where invalidity or uncoincidence is found when the received data are authenticated, processing is interrupted as a failure of mutual authentication.

In the above-described authentication process, A and B co-own a common authentication key Kab. The common authentication key Kab is distributed to a device using the enabling block key (EKB).

For example, in the example shown in FIG. 12, there may be employed the constitution in which out of A or B, the other encrypts an authentication key Kab and an enabling key block (EKB) produced by producing a decodable enabling key block (EKB) to transmit it to the other, or the constitution in which a third party

produces an enabling key bock (EKB) that can be used by both devices A and B for the devices A and B to encrypt an authentication key Kab by the enabling key block (EKB) produced for the devices A, B to distribute it.

FIGS. 13 and 14 show examples of the constitution in which an authentication key Kake common to a plurality of devices is distributed by an enabling key block (EKB). FIG. 13 shows an example in which a decodable authentication key Kake is distributed to devices 0, 1, 2, 3, and FIG. 14 shows an example in which the device 3 out of the devices 0, 1, 2, 3 is revoked to distribute a decodable authentication key to only the devices 0, 1, 2.

In the example of FIG. 13, a node key K(t)00 renewed using a node key and a leaf key in the devices 0, 1, 2, 3 is produced and distributed, by producing a decodable enabling key block (EKB), along with data (b) having an authentication key Kaka decrypted by a renewal node key K(t)00. First, the respective devices, as shown on the right side of FIG. 13, processes (decrypts) EKB to thereby obtain a renewed node key K(t)00, and then decrypts an authentication key: Enc(K(t)00, Kake) encrypted using the obtained node key K(t)00 to obtain an authentication key Kake.

In other devices 4, 5, 6, 7 ..., even if the same enabling key block (EKB) is received, the node key K(t)00 renewed by processing EKB cannot be obtained, and therefore, an authentication key can be sent to only the valid device safely.

On the other hand, the example of FIG. 14 is an example in which as the

device is, for example, revoked by leak of a key, the device 3 in a group surrounded by the dotted frame of FIG. 3 produces a decodable enabling key block (EKB) with respect to the only members of the other group, that is, the devices 0, 1, 2 for distribution. Data having (a) an enabling key block (EKB) and (b) an authentication key (Kake) shown in FIG. 14 encrypted by the node key (K(t)00) are distributed.

On the right side of FIG. 14, the decrypting procedure is shown. First, the devices 0, 1, 2 obtains an enabling node key (K(t)00) by decrypting process using a leaf key or a node key owned by itself from the received enabling key block. Next, the devices obtain an authentication Key Kake by decrypting made by K(t)00.

The devices 4, 5, 6 ... in the other group shown in FIG. 3 cannot obtain a renewal node key (K(t)00) using a leaf key and a node key owned by itself even if similar data (EKB) is received. Similarly, also in the device 3 revoked, the renewal node key (K(t)00) cannot be obtained by a leaf key and a node key owned by itself, and only the device having a valid right is able to decrypt an authentication key for use.

If distribution of an authentication key making use of EKB is used, only the valid right holder is able to distribute a decodable authentication key safely with less data quantity.

[Distribution of content key using a public key authentication and an enabling key block (EKB)]

In the following, the distribution process of the content key using a public

key authentication and an enabling key block (EKB) will be described. First, a mutual authentication method using an elliptic curve encryption of 160-bit length, which is a public key encryption system, will be described with reference to FIG. 15. In FIG. 15, ECC is used as the public key encryption system, but any system may be used as long as it is a public key encryption system similar thereto. Further, the key size need not be 160 bits. In FIG. 15, first, B produces the random number Rb of 64 bits to transmit it to A. A which received it newly produces the random number Ra of 64 bits and the random number Ak smaller than the prime number p. And, a point Av = Ak x G obtained by making a base point G, Ak times is obtained to produce an electronic signature A, Sig with respect to Ra, Rb, Av (X coordinate and Y coordinate), which is returned along with a public certificate of A to B. In Ra and Rb, X coordinate and Y coordinate of 64 bits, Av are respectively 160 bits, and therefore, an electronic signature with respect to 448 bits in total is produced.

B which received the public key certificate, Ra, Rb, Av, the electronic signature A. Sig authenticates if Rb transmitted by A is coincided with one produced by B. As a result, when coincided, an electronic signature within the public key certificate of A is authenticated by a public key of an authentication office to take out a public key of A. The electronic signature A. Sig is authenticated using a public key of A taken out.

Next, B produces the random number Bk which is smaller than the prime number p. A point Bv = Bk x G obtained by making a base point G Bk times is

obtained to produce an electronic signature B. Sig with respect to Rb, Ra, Bv (X coordinate and Y coordinate), which is returned to A along with a public key certificate of B.

A which received the public key certificate, Rb, Ra, Av, the electronic signature B. Sig of B authenticates if Ra transmitted by B is coincided with one produced by A. As a result, when coincided, an electronic signature within the public key certificate of B is authenticated by a public key of an authentication office to take out a public key of B. The electronic signature B. Sig is authenticated using a public key of B taken out. After the authentication of an electronic signature has been succeeded , A authenticates B to be valid.

Where both of them have succeeded for authentication, B computes Bk x Av (Since Bk is the random number, but Av is the point on the elliptic curve, scalar-times computation at the point on the oval curve is necessary.), and A computes Ak x Bv, and uses the lower 64 bits of the X coordinate of these points as a session key for use for thereafter communication (where a common key encryption is a common key encryption of 64 bit key length). Of course, a session key may be produced from the Y coordinate, and the coordinate need not be the lower 64 bits. In the secrete communication after mutual authentication, sometimes, the transmission data is not only encrypted by a session key but is also applied with an electronic signature.

Where in the authentication of an electronic signature or authentication of

the received data, invalidity or uncoincidence is found, processing is interrupted due to a failure of mutual authentication.

FIG. 16 shows an example of distribution process of content keys using a public key authentication and an enabling key block(EKB), First, the authentication process according to the public key system explained referring to FIG. 15 is executed between a content provider and PC. The content provider produces a decodable EKB by a reproducing apparatus which is a content key distribution destination, a node key and a leaf key owned by a recording medium to encrypt a content key E(Kcon) which executed encryption by a renewal node key and an enabling key block (EKB) by a session key Kses produced by the authentication process between PCs, which is transmitted to PC.

PC decrypts [a content key E (Kcon) which executed encryption by a renewal node key and an enabling key block (EKB)] encrypted by a session key, and thereafter transmits it to a reproducing apparatus and a recording medium.

The reproducing apparatus and the recording medium decrypt [a content key E (Kcon) which executed encryption by a renewal node key and an enabling key block (EKB)] to thereby obtain a content key Kcon.

According to the above constitution, since [a content key E (Kcon) which executed an encryption by a renewal node key and an enabling key block (EKB)] are transmitted under the condition of the authentication between a content provider and PC, for example, even in the case where a node key is leaked, positive data

transmission to a mating party is enabled.

[Distribution of a program code by using an enabling key block (EKB)]

While in the above-described example, a description has been made of a method for encrypting a content key, an authentication key or the like using an enabling key block (EKB) to distribute it, the constitution in which various program codes are distributed using an enabling key block (EKB) may be employed. That is, this is an example in which encrypted message data by EKB is used as a program code. This constitution will be described hereinafter.

FIG. 17 shows an example in which a program code is encrypted, for example, by a renewal node key of an enabling key block (EKB) to transmit it between devices. A device 1701 transmits an enabling key block (EKB) that can be decrypted by a node key and a leaf key of a device 1702 and a program code subjected to decrypting by a renewal node key contained in the enabling key block (EKB) to a device1702. The device 1702 processes the received EKB to obtain a renewal node key, and further executes decrypting of a program code by a renewal node key obtained to obtain a program code.

In the example shown in FIG. 17, further, processing by the program code obtained in the device 1702 is executed to return the result to the device 1701, and the device 1701 further continues processing on the basis of the result.

As described above, the enabling key block (EKB) and the program code subjected to decrypting processing by the renewal node key contained in the

enabling key block (EKB) are distributed whereby a program code capable of being decrypted in a specific device can be distributed to the specific device or the group shown in FIG. 3.

[Constitution for causing ICV: Integrity Check Value to correspond to a transmission content]

Next, a description will be made of the processing constitution in which for preventing falsification of a content, the integrity check value (ICV) is produced to correspond to the content, and the presence or absence of the falsification of the content is judged by computing ICV.

The integrity check value (ICV) is, for example, computed using a hash function with respect to the content, and is computed by ICV =hash (Kicv, C1, C2, ...) . Kicv is an ICV producing key. C1, C2 are information of a content, and a message authentication code (MAC) of important information of the content is used.

FIG. 18 shows a MAC value producing example using the DES encryption processing constitution. As shown in the constitution of FIG. 18, a message to be an object is divided into 8-bit units (hereinafter, the divided messages are M1, M2, ... MN). First, the initial value (hereinafter, IV) and M1 are subjected to exclusive OR (result of which is I1). Next, I1 is put into a DES encryption part to carry out encrypting using a key (hereinafter, K1) (an output is E1). Continuously, E1 and M2 are subjected to exclusive OR, output I2 of which is put into the DES

encryption part , and is encrypted using the key 1 (an output E2). Thereafter, this procedure is repeated, and the encrypting processing applied to all the messages. The last EN is a message authentication code (MAC).

The hash function is applied to the MAC value of the content and the ICV producing key to produce the integrity check value (ICV) of the content. ICV produced when a content is produced for which the fact that no falsification is present is assured is compared with ICV produced on the basis of a new content. If the same ICV is obtained, the fact that the content is not falsified is assured, and if ICV is different, judgment that falsification is present is made.

[Constitution for distributing a producing key Kicv of the check value (ICV) by EKB]

Next, the constitution in which Kiec which is an integrity check value (ICV) producing key of a content is sent by the enabling key block will be described. That is, this is an example in which encrypted message data by EKB is an integrity check value (ICV) producing key of a content.

FIG. 19 and FIG. 20 show an example in which where contents common to a plurality of devices are sent, an integrity check value producing key Kicv for authenticating the presence or absence of falsification of these contents is distributed by the enabling key block (EKB). FIG. 19 shows an example in which a decodable integrity check value producing key Kicv is distributed to devices 0, 1, 2, 3, and FIG. 20 shows an example in which the device 3 out of the devices 0, 1, 2, 3

is revoked, and a decodable integrity check value producing key Kicv is distributed to only the devices 0, 1, 2.

In the example of FIG. 19, a node key K(t)00 renewed using a node key and a leaf key owned by the devices 0, 1, 2, 3 along with data (b) having a check value producing key Kicv encrypted by a renewal node key K(t)00 are distributed by producing a decodable enabling key block (EKB). As shown on the right side in FIG. 19, the respective devices first process (decrypts) EKB to thereby obtain a node key K(t)00 renewed, and subsequently decrypt a check value producing key : Enc(K(t)00, Kicv) encrypted using the obtained node key K(t)00 to obtain a check value producing key Kicv.

Since other devices 4, 5, 6, 7 ... cannot obtain a node key K(t)00 renewed by processing EKB by a node key and a leaf key owned by itself even if the same enabling key block (EKB) is received, the check value producing key can be sent to only valid device safely.

On the other hand, the example of FIG. 20 is an example in which as the device is, for example, revoked by leak of a key, in a group surrounded by the dotted frame of FIG. 3. produces a decodable enabling key block (EKB) with respect to the only members of the other group, that is, the devices 0, 1, 2 for distribution. Data having (a) an enabling key block (EKB) and (b) a check value producing key (Kicv) shown in FIG. 20 encrypted by the node key (K(t)00) are distributed.

On the right side of FIG. 20, the decrypting procedure is shown. First, the devices 0, 1, 2 obtain a renewal node key (K(t)00) by decrypting process using a leaf key or a node key owned by itself from the received enabling key block. Next, the devices obtain a check value producing key Kicv by decrypting made by K(t)00.

The devices 4, 5, 6 ... in the other group shown in FIG. 3 cannot obtain a renewal node key (K(t)00) using a leaf key and a node key owned by itself even if similar data (EKB) is received. Similarly, also in the device 3 revoked, the renewal node key (K(t)00) cannot be obtained by a leaf key and a node key owned by itself, and only the device having a valid right is able to decrypt an authentication key for use.

If distribution of a check value reproducing key making use of EKB is used, only the valid right holder is able to distribute a decodable check value producing key safely with less data quantity.

By using the integrity check value (ICV) of contents as described above, it is possible to eliminate invalid copies of EKB and encrypted contents. It is supposed that for example, as shown in FIGS. 21A and 21B, there is a medium 1 in which a content C1 and a content C2 are stored along with an enabling key block (EKB) capable of obtaining content keys, which is copied to a medium 2 without modification. It is possible to copy EKB and encrypted contents, which can be used in a device capable of decrypting EKB.

There is provided a constitution in which as shown in FIG. 21B, integrity

check values (ICV (C1, C2)) are stored corresponding to contents properly stored in the respective media. The (ICV (C1, C2)) shows ICV = hash (Kicv, C1, C2) which is an integrity check value of contents computed using the hash function in the content C1 and the content C2. In the constitution of FIG. 21B, a content 1 and a content 2 are properly stored in the medium 1, and integrity check values (ICV (C1, C2)) produced on the basis of the content C1 and the content C2 are stored. Further, a content 1is properly stored in the medium 2, and an integrity check values (ICV (C1)) produced on the basis of the content C1 is stored therein. In this constitution, Assume that (EKB, content 2) stored in the medium 1is copied to the medium 2, when in the medium 2, a content check value is newly produced, ICV (C1, C2) are to be produced, so that it becomes obvious that different from Kicv (C1) stored in the medium, falsifying of contents and storing of new contents due to the invalid copy are executed. In the device for reproducing media, ICV checking is executed in the step previous to the reproducing step, and judgment is made of coincidence between the produced ICV and the stored ICV, if not coincident, the constitution in which reproducing is not executed is provided to enable prevention of reproducing contents copied invalidly.

Furthermore, there can be provided the constitution in which for enhancing safety, the integrity check value (ICV) of contents is rewritten to produce them on the basis of data including a counter. That is, this constitution is to make computation by ICV = hash (Kicv, counter + 1, C1, C2, ...). Here, a counter

(counter + 1) is set as a value in which one increment is made every rewriting. It is necessary to have a constitution in which a counter value is stored in a secure memory.

Further, in the constitution in which the integrity check value (ICV) of contents is cannot be stored in the same medium as contents, the integrity check value (ICV) of contents is stored in a separate medium.

For example, where contents are stored in media which take no measures to prevent copies such as a read only memory or normal MO, there is the possibility that when the integrity check value (ICV) is stored in the same medium, rewriting of the ICV is done by an invalid user, failing to maintain the safety of ICV. In such a case, there can be provided the constitution in which ICV is stored in a safety medium on a host machine, and ICV is used for copy control (for example, check-in / check-out, move), to thereby enable safe management of ICV and checking of falsification of contents.

The above constitution is shown in FIG. 22. In FIG. 22, contents are stored in a medium 2201 which takes no measures for preventing copying such as read only media or normal MO, and the integrity check values (ICV) in connection with these contents are stored in a safe media 2202 on a host machine to which a user is not allowed to get access to prevent invalid rewriting of the integrity check value (ICV) by a user. If, as such a constitution as described above, for example, employment is made of a constitution in which when a device on which a media

2201 is mounted executes reproducing of the media 2201, a PC or a server which is a host machine executes checking of ICV to judge the propriety of reproducing, reproducing of invalid copy contents or falsified contents can be prevented.

[Category classification of a hierarchical tree structure]

A description has been made of the constitution in which an encrypted key is constituted as a hierarchical tree structure shown in FIG. 3 such as a root key, a node key, a leaf key, etc., and a content key, an authentication key, an ICV producing key or a program code, data or the like are encrypted along with an enabling key block and distributed, but a description will be made hereinafter of the constitution in which a hierarchical tree structure which defines a node key or the like is classified every category of devices to execute efficient key renewing process, encrypted key distribution, and data distribution.

FIG. 23 shows one example of classification of category of a hierarchical tree structure. In FIG. 23, a root key Kroot 2301 is set on the uppermost stage of the hierarchical tree structure, a node key 2302 is set in the intermediate stage, and a leaf key 2303 is set in the lowest stage. Each device holds individual leaf keys, and a series of node keys from a leaf key to a root key, and a root key.

Here, as one example, nodes from the uppermost stage to the M stage is set as a category node 2304. That is, each of nodes on the M stage is set as a device setting node of a specific category. Nodes and leaves lower than the M+1 stage are taken as nodes and leaves in connection with devices contained in the category

thereof with one node in the M stage as a top.

For example, a category [Memory stick (trademark)] is set to one node 2305 in the M stage of FIG. 23, and nodes and leaves provided lower than the node 2305 are set as category-exclusive use nodes or leaves containing various devices using the memory stick. That is, those below the node 2305 are defined as the gathering of nodes and leaves associated with device defined in the category of the memory stick.

Further, a stage at a level below several stages from the M stage can be set as a sub-category node 2306. For example, a node of [Reproducing exclusive-use unit] is set as a sub-category node contained in the category of the device using the memory stick in a node two stages below a category [memory stick] node 2305 as shown in the figure. Further, a node 2307of a telephone with a music reproducing function contained in the category of the reproducing exclusive-use unit below the node 2306 of the reproducing exclusive-use unit as a sub-category node, and a [PHS] node 2308 and a [Portable telephone] node 2309 contained in the category of the telephone with a music reproducing function can be set therebelow.

Further, the category and sub-categories can be set not only at the kind of devices, but also at nodes managed independently, for example, makers, a content provider, a settlement organization or the like, that is, at suitable units such as processing unit, jurisdiction unit, or service providing unit (these will be generally called entity). For example, if one category node is set as a game machine XYZ

exclusive-use top node sold by game machine makers, a node key and a leaf key in the lower stage below the top node can be stored in the game machine XYZ sold by makers for sales, after which distribution of encrypted contents, or distribution of various keys, and renewal processing are distributed producing an enabling key block (EKB) constituted by node keys and leaf keys below the top node key, and data that can be utilized merely for the devices below the top node can be distributed.

The constitution can be provided in which the node below one node as a top is set as an associated node of the category or sub-categories defined, whereby makers, a content provider or the controlling one top node in the category stage or sub-category stage independently produces an enabling key block with the node as a top to distribute it to the devices belonging to those below the top node, and key renewing can be executed without affecting at all on the devices belonging to the nodes of other categories not belonging to the top node.

[Key distributing constitution by simplified EKB (1)]

For example, in the tree structure of FIG. 3 described previously, where for example, a content key is addressed to a predetermined device (leaf), a decodable enabling key block (EKB) is produced and provided using a leaf key and a node key owned by a key distributing device. For example, in a tree structure shown in FIG. 24A, where a key, for example, a content key is transmitted to devices a, g, j constituting a leaf, a decodable enabling key block (EKB) is produced in the nodes

a, g, j and distributed.

It is contemplated that for example, a content key K(t)con is subjected to encrypting processing by a renewal root key K(t)root to distribute it along with EKB. In this case, the devices a, g, j execute processing of EKB using a leaf key and a node key shown in FIG. 24B to obtain K(t)root, and execute decrypting process of a content key K(t)con by the obtained renewal root key K(t)root to obtain a content key.

The constitution of the enabling key block (ERK) provided in this case is as shown in FIG. 25. The enabling key block (ERK) shown in FIG. 25 is constituted in accordance with the format of the enabling key block (EKB) explained previously with reference to FIG. 6, has a tag corresponding to data (encrypted key).The tag is 0, if data is present in the directions of left (L) and right (R), and is 1 if not, as previously explained with reference to FIGS. 7A to 7C.

The device which received the enabling key block (EKB) sequentially executes decrypting process of encrypted keys on the basis of an encrypted key of the enabling key block (EKB) and the tag to obtain a renewal key of an upper node. As shown in FIG. 25, in the enabling key block (EKB), the more the number of stages (depth) from a root to a leaf, the quantity of depths increases. The number of stages (depth) increases according to the number of devices (leaf), and where there are many numbers of devices to be a distributing destination of keys, the data quantity of EKB further increases.

The constitution in which the reduction of data quantity of the enabling key block (EKB) as described is enabled will be described below. FIGS. 26A and 26B show an example in which the enabling key block (EKB) is simplified according to the key distribution device.

It is assumed that similarly to FIG. 25, a key, for example, a content key is transmitted to devices a, g, j constituting a leaf. As shown in FIG. 26A, a tree constituted merely by a key distributing device is constructed. In this case, a tree constitution of FIG. 26B is constructed as a new tree constitution based on the constitution shown in FIG. 24B. No branch is present from Kroot to Kj, but only one branch will suffice, and , from K root to Ka and Kg, a tree of FIG. 26A having a 2-branch constitution is constructed merely by constituting a branch point at K0.

As shown in FIG. 26A, a simplified tree having only K0 as a node is produced. The enabling key block (EKB) for the renewal key distribution is produced on the basis of these simplified trees. The tree shown in FIG. 26 (a) is a re-constructed hierarchical tree re-constructed by selecting a pass constituting a 2-branch type tree with a decodable terminal node or leaf as the lowest stage to omit unnecessary nodes. The enabling key block (EKB) for distributing a renewal key is constituted on the basis of only the key corresponding to a node or a leaf of the re-constructed hierarchical tree.

The enabling key block (EKR) described previously with reference to FIG. 25 stores data having all keys from leaf a, g, j to Kroot, but the simplified EKB

stores encrypted data with respect to only the nodes constituting the simplified tree. As shown in FIG. 26B, the tag has a 3-bit constitution. A first bit and a second bit have meaning similar to that of the example of FIG. 25, in which if data are present in the directions of left (L) and right (R), it indicates 0, and if not, 1. A third bit is a bit for indicating that whether or not an encrypted key is contained in EKB, and if data is stored, 1 appears, and if not, 0 appears.

An enabling key block (EKB) provided for a device (leaf) stored in a data communication network or a memory medium is considerably reduced in data quantity as shown in FIG. 26B, as compared with the constitution shown in FIG. 25. Each deice which received the enabling key block (EKB) shown in FIGS. 26A and 26B sequentially decrypts only data in a portion where 1 is stored in the third bit of the tag to enable realization of decrypting of a predetermined encrypted key. For example, the device a decrypt Enc(Ka, K(t)0) by a leaf key Ka to obtain a node key K(t)0, and decrypts encrypted data Enc(K(t)0, K(t)root) by a node key K(t)0 to obtain K(t)root. The device j decrypts encrypted data Enc(Kj, K(t)root) by a leaf key Kj to obtain K(t)root.

The enabling key block (EKB) is produced using only the keys of leaf and node which constructs a simplified new tree constitution constituted merely by the device of the distributing destination to constitute a constructed tree to thereby enable producing an enabling key block (EKR) with less data quantity, and the data distribution of the enabling key block (EKB) can be executed efficiently.

[Key distributing constitution by simplified EKB (2)]

The constitution will be described in which the enabling key block (EKB) produced on the basis of the simplified tree shown in FIGS. 26A and 26B are further simplified to enable reduction of data quantity and efficient processing.

The constitution described with reference to FIGS. 26A and 26B is the re-constructed hierarchical tree reconstructed by selecting a pass constituting a 2-branch type tree with the decodable terminal node or leaf as the lowermost stage to omit unnecessary nodes. The enabling key block (EKB) for distributing a renewal key is constituted on the basis of only the key corresponding to a node or a leaf of the re-constructed hierarchical tree.

The re-constructed hierarchical tree shown in FIG. 26A distributes the enabling key block (EKB) shown in FIG. 26B to enable obtaining the renewal root key Kroot in the leaf a, g, j. In processing the enabling key block (EKB) of FIG. 26B , the leaf j is possible to obtain the root key ((T)root by one time decrypting process of Enc(Kj, K(t)root). However, the leaf a, g obtain K(t)0 by decrypting of Enc(Kg, K(t)0), and thereafter further executes decrypting process of Enc(K(t)0, K(t)root) to obtain a root key K(t)root. That is, the leaf a, g are necessary to execute decrypting process twice.

In the reconstructed re-constructed hierarchical tree of FIGS. 26A and 26B, where the node K0 executes its own control as a control node of lower leaf a, g, for example, where executes control of lower leaf as a sub-root node described later, it

is effective in a sense of confirming that the leaf a, g obtain a renewal key, but, where the encode K0 does not carry out control of the lower leaf, or where even if the control is carried out, distribution of a renewal key from the upper node is allowed, the re-constructed hierarchical tree shown in FIG. 26A may be further simplified to omit the key of node K0 to produce the enabling key block (EKB) for distribution.

FIGS. 27A and 27B show the constitution of the enabling key block (EKB) as described above. It is assumed similarly to FIGS. 26A and 26B that a key, for example, a content key is transmitted to the devices a, g, j constituting a leaf. As shown in FIG. 27A, a tree is constructed in which a root Kroot and leaf a, g, j are connected directly.

As shown in FIG. 27A, a simplified tree having a node K0 omitted from the re-constructed hierarchical tree shown in FIG. 26A is produced. The enabling key block (EKB) for distributing a renewal key is produced on the basis of these simplified trees. The tree shown in FIG. 27A is a re-constructed hierarchical tree re-constructed merely by a pass for directly connecting a decodable leaf and a root, The enabling key block (EKB) for distributing a renewal key is constituted on the basis of a key corresponding to a leaf of the re-constructed hierarchical tree.

Although the example of FIG. 27A is an example of the constitution in which a terminal is a leaf, it is possible, in a case of distributing keys to the uppermost node or a plurality of middle and lower nodes, to produce the enabling key block

(EKB) on the basis of the simplified tree in which the uppermost node and the middle and lower nodes are directly connected to execute key distribution. As described above, the re-constructed hierarchical tree has a constitution in which a top node constituting the simplified tree, a terminal node or leaf constituting the simplified tree are directly connected. In the simplified tree, it is possible to constitute it as a tree having not only two branches from the top node but a multi-branch not less than three according to the number of distribution nodes or leaves.

The enabling key block (EKB) described previously with reference to FIG. 25 has the constitution in which data having all keys from each leaf a, g, j to Kroot encrypted are stored, and the enabling key block (EKB) stores K0 as a common node of leaf keys a, g of leaf a, g, j, and a root key, but the enabling key block (EKB) based on the simplified hierarchical tree shown in FIG. 27A omits a key of node K0, and therefore, the enabling key block (EKB) with less data quantity is obtained, as shown in FIG. 27B.

The enabling key block (EKB) shown in FIG. 27B has a tag of 3 bits similarly to the enabling key block (EKB) shown in FIG. 26B. In a first and a second bits, if data are present in the directions of left (L) and right (R), it indicates 0, and if not, 1. A third bit is a bit for indicating whether or not an encrypted key is stored within EKB, and where data is stored, 1 appears, and if not, 0 appears.

In the enabling key block (EKB) of FIG. 27B, each leaf a, g, j is possible to

obtain a root key K(t)root by one-time decrypting process of Enc(Ka, K(t)root), or Enc(Kg, K(t)root) Enc(Kj, K(t)root).

The enabling key block (EKB) produced on the basis of the tree having the constitution in which the uppermost node of the simplified re-constructed hierarchical tree, the terminal node constituting a tree or a leaf are directly connected is constituted on the basis of only the key corresponding to the top node and the terminal node or the leaf of the re-constructed hierarchical tree.

A simplified new tree constitution constituted merely by a device of distributing destination, and the enabling key block (EKB) is produced using only the leaf constituting the constructed tree or only the key of node common to a leaf, as in the enabling key block (EKB) described with reference to FIGS. 26A and 26B or FIGS. 27A and 27B, to thereby make it possible to produce the enabling key block (EKB) with less data quantity and to effectively execute data distribution of the enabling key block (EKB).

The simplified hierarchical tree constitution can be utilized effectively particularly in the EKB control constitution in entity unit described later. The entity is a gathering block of a plurality of nodes or leaf selected from a node or a leaf constituting a tree constitution as a key distribution constitution. The entity is set as the gathering set according to the kind of devices, or set as the gathering of a variety of forms such as a processing unit, a control unit, or a service providing unit having a common point such as control units of a device providing maker, a content

provider, a settlement organization or the like. Devices classified into categories are gathered in a single entity. For example, a simplified tree similar to that described above is re-constructed by top node (sub-roots) of a plurality of entities to produce EKB to thereby make it possible to produce and distribute the decodable simplified enabling key block (EKB) in the device belonging to the selected entity. The control constitution of the entity unit will be described in detail later.

Such an enabling key block (EKB) as described above can be constituted to be stored in information recording medium such as an optical disk, DVD or the like. For example, there can be provided the constitution in which an information recording medium, in which message data such as contents encrypted by a renewal node key is stored in the enabling key bock (EKB) containing data part constituted by the aforementioned encrypted key data and a tag part as position discrimination date in the hierarchical tree structure of encrypted key data, is provided for each device. The device sequentially extracts and decrypts encrypted key data contained in the enabling key block (EKB) in accordance with the discrimination data of the tag part. Of course, there can be employed the constitution in which the enabling key block (EKB) is distributed through a network such as an internet.

[EKB control constitution of entity unit]

Next, a description will be made of the constitution in which a node or a leaf constituting a tree constitution as a key distribution constitution is controlled by a block as the gathering of a plurality of nodes or leaves. The block as the gathering

of a plurality of nodes or leaves will be hereinafter called an entity. The entity is set as the gathering set according to the kind of devices or as the gathering of various forms such as a processing unit, a jurisdiction unit or a service providing unit having a common point such as device providing makers, a content provider or a settlement organization.

The entity will be described with reference to FIGS. 28A to 28C. FIG. 28A is a view for explaining the control constitution in entity unit of a tree. One entity is shown as a triangle in the figure, for example, a plurality of nodes are contained in 1 entity 2701. FIG. 28B shows the node constitution within the 1 entity. The 1 entity is constituted by a plurality of 2-branch type trees as one node as a top. The top node 2702 of the entity will be hereinafter called a sub-root.

The terminal of the tree is constituted by a leaf as shown in FIG. 28C, that is, a device. The device belongs to any entity constituted by a tree with a plurality of device as a leaf and having a top node 2702 which is a sub-root.

As will be understood from FIG. 28A, the entity has a hierarchical structure. This hierarchical structure will be described with reference to FIGS. 29A to 29C.

FIG. 29A is a view for explaining the hierarchical structure in a simplified form. Entities A01 to Ann are constituted in the stage several stages below Kroot, entities B01 to Bnk are set below the entities A1 to An, and entities C1 to Cnq are set thereunder. Each entity has a tree shape constituted by plural stages of nodes and leaves, as shown in FIGS. 29B and 29C.

For example, the constitution of the entity Bnk has a plurality of nodes to a terminal node 2812 with a sub-root 2811 as a top node. This entity has a discriminator Bnk, and the entity Bnk independently executes node key control corresponding to the node within the entity Bnk to thereby execute control of a lower (child) entity set with the terminal node 2812 as a top. On the other hand, the entity Bnk is under the (host) entity Ann having the sub-node as a terminal node 2811.

The constitution of an entity Cn3 has a terminal node 2852 which is each device with a sub-root 2851 as a top node, and a plurality of nodes and leaves to a leaf in this case, as shown in FIG. 29C. This entity has a discriminator Cn3, the entity Cn3 independently executes control of a node key and a leaf key corresponding to the node and leaf within the entity Cn3 to thereby execute control of a leaf (device) corresponding to the terminal node 2852. On the other hand, the entity Cn3 is under the (host) entity Bn2 having the sub-root 2851 as a terminal node. The key control in each entity is, for example, key renewing process, revoke process and the like, which will be described in detail later.

In a device which is a leaf of the lowest entity are stored a node key of each node and a leaf key positioned in a pass from a leaf key of entity to which the device belongs to a sub-root node which is a top node of entity to which itself belongs. For example, the device of the terminal node 2852 stores keys from the terminal node (leaf) 2852 to the sub-root node 2851.

The constitution of the entity will be further described with reference to FIGS. 30A and 30B. The entity is able to have a tree structure constituted by a variety of stage numbers. The stage number, that is, the depth can be set according to the number of child entities corresponding to the terminal node controlled by the entity, or the device number as a leaf.

The detail of the constitution of host and child entities as shown in FIG. 30A is as shown in FIG. 30B, The root entity is an entity in the uppermost stage having a root key. Entities A, B, C are set as a plurality of child entities in the terminal node of the root entity, and an entity D is set as a child entity of entity C. An entity C2901 has not less than one node of the terminal node as a sub-node 2950, and where entities controlled by itself are increased, an entity C'2902 having plural stages of trees is newly installed with a reserve node 2950 as a top node to thereby increase control terminal nodes 2970, and a child entity increased can be added to the control terminal node.

The reserve node will be further described with reference to FIG. 31. Entity A, 3011 has child entities B, C, D ... to be controlled, and has one reserve node 3021. Where child entities to be controlled are further increased, a child entity A', 3012 under the own control is set to the reserve node 3021, and child entities F, G to be controlled can be further set to the terminal node of the child entity A', 3012. Also in the child entity A', 3012 under the own control, at least one of the terminal nodes is set as a reserve node 3022 whereby a child entity A"3013 is further set to

further increase the control entities. One or more reserve nodes are secured also in the terminal node of the child entity A"3013. Such a reserve node holding constitution as described is employed whereby the child entities under a certain entity can be increased endlessly. With respect to the reserve entity, not only one terminal node but a plurality of nodes may be set.

In the respective entities, the enabling key block (EKB) is constituted in entity unit, and key renewing and revoke processing are to be executed in entity unit. As shown in FIG. 31, the enabling key block (EKB) of individual entity is set to a plurality of entities A, A', A", but these can be collectively controlled, for example, by device makers who controls the entities A, A', A" in common.

[Registration process of new entities]

Next, the registration process of new entities will be described. FIG. 32 shows a registration processing sequence. A description will be made in accordance with the sequence in FIG. 32. A new (child) entity(N-En) newly added during the constitution of a tree executes requesting of new registration to a host entity (P-En). Each entity holds a public key in accordance with a public key encryption system, and a new entity sends own public key to the host entity (P-En) when registration request is made.

The host entity (P-En) which received the registration request transfers a public key of the new a (child) entity received to a certificate authority (CA) and receives a public key of the new (child) entity (N-En) to which a signature of CA is

added. These procedures are carried out as a procedure for mutual authentication between the host entity (P-En) and the new (child) entity (N-En).

When the authentication of the new registration requesting entity is terminated, the host entity (P-En) grants the registration of the new (child) entity (N-En) to transmit a node key of the new (child) entity (N-En) to the new (child) entity (N-En) . This node key is one node key of the terminal node of the host entity (P-En) which corresponds to a top node of the new (child) entity (N-En), that is, a sub-root key.

When the transmission of node key is finished, the new (child) entity (N-En) constructs the tree constitution of the new (child) entity (N-En), sets a sub-root key of a top node received to a top of the constructed tree, and sets node and leaf keys to produce an enabling key block (EKB) within the entity. The enabling key block (EKB) within one entity is called a sub-EKB.

On the other hand, the host entity (P-En) produces the sub-EKB within the host entity(P-En) to which is added a terminal node to be enabled by the addition of the new (child) entity (N-En).

When the sub-EKB constituted by a node key and a leaf key within the new (child) entity (N-En) is produced, the new (child) entity (N-En) transmits it to the host entity (P-En).

The host entity (P-En) which receives the sub-EKB from the new (child) entity (N-En) transmits the received sub-EKB and a renewal sub-EKB of the host

entity (P-En) to a key distribute center (KDC) .

The key distribute center (KDC) is able to produce various EKBs, that is, EKB that can be decrypted merely by a specific entity or device on the basis of sub-EKBs of all entities. EKB to which such a decodable entity or device is set is distributed, for example, to a content provider, who encrypts a content key on the basis of EKB to distribute it through a network or store it in a recording medium, thus enabling distribution of a content that can be used merely by a specific device.

The registration processing with respect to the key distribute center (KDC) of the sub-EKB of the new entity is not limited to a method for sequentially transferring the sub-EKB through the host entity, but there can be also employed the constitution which executes the processing for registering the sub-EKB in the key distribute center (KDC) directly from the new registration entity without the intervention of the host entity.

The correspondence of the host entity to the child entity to be newly added to the host entity will be described with reference to FIG. 33. One terminal node 3201 of the host entity is distributed as a top node of the newly added entity to the child entity whereby the child entity is added as an entity under the control of the host entity. The entity under the control of the host entity termed herein, which will be described later, also includes meaning of the constitution in which the revoke processing of the child entity can be executed by the host entity.

As shown in FIG. 33, when a new entity is set to the host entity, one node

3201 of a terminal node which is a leaf of the host entity and a top node 3202 of the newly added entity are set as equal nodes. That is, one terminal node which is one leaf of the host node is set as a sub-root of the newly added entity. By being so set, the newly added entity is enabled under the whole tree constitution.

FIGS. 34A and 34B show an example of a renewal EKB produced by the host entity when the newly added entity is set. FIG. 34A shows an example of a sub-EKB produced by the host entity when a new entity added terminal node (node 100) 3303 is applied to the newly added entity, in the constitution shown in FIG. 34A which has a terminal node (node 000) 3301 which has been effectively present and a terminal node (node 001) node 3302.

The sub-EKB has the constitution as shown in FIG. 34B. There are a host node key encrypted by a terminal node which has been effectively present, a further host node key encrypted by the host node key, ... and a sub-root key. Similarly to FIG. 34B, each entity has and controls EKB constituted to have a host node encrypted by an effective terminal node or leaf key, encrypts a further host node key by the host node key, and an encrypted data to a sub-root sequentially being increased in depth.

[Revoke processing under the control of entity]

Next, a description will be made of the revoke processing of a device or an entity in the constitution in which the key distribution tree constitution is controlled as an entity unit. In previous FIGS. 3 and 4, a description has been made of the

processing for distributing an enabling key block (EKB) in which only the specific device out of the whole tree constitution is decodable, and the revoked device is undecodable. The revoke processing described in FIGS. 3 and 4 is the processing for revoking a device which is a specific leaf out of the whole tree, but the constitution by entity control of the tree is possible to execute the revoke processing every entity.

A description will be made hereinafter of the revoke processing in the constitution under the entity control with reference to FIGS. 35A to 35D and drawings continuous thereto. FIGS. 35A to 35D is a view for explaining the revoke processing of a device by an entity which controls an entity in the lowest stage, out of entities constituting a tree, that is, an entity controlling individual devices.

FIG. 35A shows the key distribution tree structure under the control of entity. A root node is set to the uppermost part of the tree, and entities A01 to Ann, entities B01 to Bnk below the previous entities, and entities C1 to cn in the lowest stage are constituted. In the lowest entity, the terminal node (leaf) is individual devices, for example, a recording and reproducing unit, a reproducing exclusive-use unit or the like.

The revoke processing is independently in each entity. For example, in the entities C1 to Cn in the lowest stage, the revoke processing of a device of a leaf is executed. FIG. 35B shows the tree constitution of an entity Cn, 3430 which is one of the entities in the lowest stage. The entity Cn, 3430 has a top node 3431, and a

leaf which is a terminal node has a plurality of devices.

Assume that a device to be revoked, for example, a device 3432 is present in a leaf, the entity Cn, 3430 produces an enabling key block (sub-EKB) constituted by a node key and a leaf key in the independently renewed entity Cn. This enabling key block is a key block constituted by an encrypted key that cannot be decrypted in the revoke key in the revoke device 3432 but that can be decrypted by only the device constituting other leaf. A controller of the entity Cn produce it as a renewed sub-EKB. Concretely, the block, which comprises an encrypted key which renews node keys of nodes 3431, 3434, and 3435constituting a pass associated with a sub-root to a revoke device 3432, and can decrypt the renewal key only in a leaf device other than the revoke device 3432. This processing corresponds to the processing in which a root key is replaced by a sub-root which is a top key of entity, in the revoke processing constitution described in FIGS. 3 and 4.

The enabling key block (sub-EKB) renewed by the entity Cn, 3430 through the revoke processing is transmitted to the host entity. In this case, the host entity is an entity Bnk, 3420, and an entity having a top node 3431 of the entity Cn, 3430 as a terminal node.

The entity Bnk, 3420, when receives the enabling key block (sub-EKB) from the child entity Cn, 3430, sets the terminal node 3431 of the entity Bnk, 3420 corresponding to the top node 3431 of the entity Cnk, 3430 contained in the key block to a key renewed in the child entity Cn, 3430, and executes the renewal

processing of sub-EKB of own entity Bnk, 3420. FIG. 35C shows the tree of entity Bnk, 3420. In the entity Bnk, 3420, a node key to be renewed is a node key on a pass from the sub-root 3421 in FIG. 35C to the terminal node 3431 constituting an entity containing a revoke device. That is, node keys of the nodes 3421, 3424, 3425 constituting a pass associated with the node 3431 of the entity transmitted from the renewal sub-EKB are to be renewed. These node keys of nodes are renewed to produce a new renewal sub-EKB of the entity Bnk, 3420.

Further, the enabling key block (sub-EKB) renewed by the entity Bnk, 3420 is transmitted to the host entity. In this case, the host entity is the entity Ann, 3410, and an entity having a top node 3421 of the entity Bnk, 3420 as a terminal node.

The entity Ann, 3410, when receives the enabling key block (sub-EKB) from the child entity Bnk, 3420, sets the terminal node 3421 of the entity Ann, 3410 corresponding to the top node 3421 of the entity Bnk, 3420 contained in the key block to a key renewed in the child entity Bnk, 3420, and executes the renewal processing of sub-EKB of own entity Ann, 3410. FIG. 35D shows the tree of entity Ann, 3410. In the entity Ann, 3410, node keys to be renewed are node keys 3411, 3414, 3415 on a pass from the sub-root 3411 in FIG. 35D to the terminal node 3421 constituting an entity containing a revoke device. These node keys of nodes are renewed to produce a new renewal sub-EKB of the entity Ann, 3410.

These processes sequentially execute in the host entity to the root entity described in FIG. 30B. The revoke processing of devices is completed by a series of

processes as described. The sub-EKB renewed in the entity is finally transmitted to the key distribute center (KDC) and stored therein. The key distribute center (KDC) produces various EKBs on the basis of the renewal sub-EKB of all entities. The renewal EKB is an encrypted key block that cannot be decrypted by the device revoked.

FIG. 36 shows a sequence of revoke process of the device. The processing procedure will be described with reference to the sequence figure of FIG. 36. First, the device control entity (D-En) in the lowest stage of the tree constitution carries out key renewing necessary for revoking a leaf to be revoked in the device control entity (D-En) to produce a new sub-EKB of the device control entity (D-En). The sub-EKB is sent to the host entity. The host entity (P1- En) which received the renewal sub-EKB (D) produces a renewal sub-EKB (P1) in which a terminal node key corresponding to a renewal top node of the renewal sub-EKB (D) is renewed and node keys on a pass from the terminal node to the sub-root. These processes are sequentially executed in the host entity, and all sub-EKBs finally renewed are stored and controlled by the key distribute center (KDC).

FIGS. 37A and 37B show an example of an enabling key block (EKB) to be produced as a result that the host entity carries out renewal processing by the revoke processing of a device.

FIGS. 37A and 37B are views each for explaining an example of EKB produced in the host entity which received renewal sub-EKB from the child entity

containing a revoke device, in the constitution shown in FIG. 37A. A top node of the child entity containing the revoke device corresponds to a terminal node (node 100) 3601 of the host entity.

The host entity renews node keys which are present in a pass from the sub-root of the host entity to the terminal node (node 100) 3601 to produce a new renewed sub-EKB. The renewal sub-EKB is as shown in FIG. 37B. The renewed key is shown with the underline and ['] attached thereto. The node keys on a pass from the renewed terminal node to the sub-rot are renewed to obtain a renewal sub-EKB in its entity.

Next, processing where an object subjected to revoking is an entity, that is, revoke processing of entity, will be described.

FIG. 38A shows the key distribution tree structure by entity control. A root node is set to the uppermost part of the tree, and entities A01 to Ann are constituted several stages thereunder, entities B01 to Bnk are constituted in the stage lower than the former, and entities C1 to cn are constituted in the stage lower than the further stage are constituted. In the lowest entity, the terminal node (leaf) is individual devices, for example, such as recording and reproducing unit, a reproducing exclusive-use unit or the like.

Now, a description is made of the case where the revoke processing is carried out with respect to the entity Cn, 3730. The entity Cn, 3730 in the lowest stage has the constitution in which a top node 3431 is provided, and a plurality of devices are

provided on a leaf which is a terminal node, as shown in FIG. 38B.

The revoking of the entity Cn, 3730 enables collective revoke of all devices belonging to the entity Cn, 3730 from the tree structure. The revoke processing of the entity cn, 3730 is executed in the entity Bnk, 3720 which is the host entity of the entity Cn, 3730. The entity Bnk, 3720 is an entity having the top node 3731 of the entity Cn, 3730 as a terminal node.

Where revoking of the child entity Cn, 3730 is executed, the entity Bnk, 3720 renews a terminal node 3731 of the entity Bnk, 3720 corresponding to the top node 3731 of the entity Cnk, 3730, and further carries out renewing of node keys on a pass from the revoke entity 3730 to the sub-root of the entity Bnk, 3720 to produce an enabling key block to produce a renewal sub-EKB. The node key to be renewed is a node key on a pass from the sub-root 3721 shown in FIG. 38C to a top node of a revoke entity. That is, nodes 3721, 3724, 3725 and 3731 are objects to be renewed. These node keys of nodes are renewed to produce new renewal sub-EKB of the entity Bnk, 3720.

Alternatively, where revoking of the child entity Cn, 3730 is executed, the entity Bnk, 3720 does not renew the terminal node 3731 of the entity Bnk, 3720 corresponding to the top node 3731 of the entity Cnk, 3730 but renews a node key except the terminal node 3731 on the pass from the revoke entity 3730 to the sub-root of the entity Bnk, 3720 to produce an enabling key block to produce a renewal sub-EKB.

Further, the enabling key block (sub-EKB) renewed by the entity Bnk, 3720 is transmitted to the host entity. In this case, the host entity is an entity Ann, 3710, which is an entity having a top node 3721 of the entity Bnk, 3720 as a terminal node.

When an enabling key bock (sub-EKB) is received from the child entity Bnk, 3720, the entity Ann, 3710 sets the terminal node 3721 of the entity Ann, 3710 corresponding to the top node 3721 of the entity Bnk, 3720 contained in the key block to a key renewed in the child entity Bnk, 3720 to execute renewal processing of the sub-EKB of the own entity Ann, 3710. FIG. 38D shows the tree constitution of the entity Ann, 3710. In the entity Ann, 3710, the node key to be renewed is a node key of each node 3711, 3714, 3715 constituting a pass from the sub-root 3711 of FIG. 38D to the node 3721 of the entity having transmitted the renewal sub-EKB. These node keys of the nodes are renewed to produce a new renewal sub-EKB of the entity Ann, 3710.

These processes are sequentially executed in the host entity to execute it to the root entity described with reference to FIG. 30B. The revoke processing is completed by a series of processes. The sub-EKB renewed in the respective entity is finally transmitted to the key distribute center (KDC) and stored. The key distribute center KDC produces various EKBs on the basis of the renewal sub-EKB of all entities. The renewal EKB is an encrypted key block that cannot be decrypted by the device belonging to the entity revoked.

FIG. 39 shows a sequence of revoke process of the entity. The processing procedure will be described with reference to the sequence figure of FIG. 39. First, the entity control entity (E-En) for revoking the entity carries out key renewing necessary for revoking a terminal node to be revoked in the entity control entity (E-En) to produce a new sub-EKB of the entity control entity (E-En) . The sub-EKB is sent to the host entity. The host entity (P1- En) which received the renewal sub-EKB (E) produces a renewal sub-EKB (P1) in which a terminal node key corresponding to a renewal top node of the renewal sub-EKB (P1) is renewed and node keys on a pass from the terminal node to the sub-root are renewed. These processes are sequentially executed in the host entity, and all sub-EKB finally renewed are stored and controlled by the key distribute center (KDC). The key distribute center (KDC) produces various EKB on the basis of the renewal EKB of all entities. The renewal EKB is an encrypted key block that cannot be decrypted by a device belonging to the entity revoked.

FIG. 40 is a view for explaining the correspondence of the child entity revoked to the host entity which carried out revoking. A terminal node 3901 of the host entity is renewed by revoking the entity, and a new sub-EKB is produced by renewing of node keys which are present in a pass from the terminal node 3901 to the sub-root in the tree of the host entity. As a result, the node key of the top node 3902 of the child entity revoked is not coincided with the node key of the terminal node 3901 of the host entity. EKB produced by the key distribute center (KDC)

after revoking of the entity is to be produced on the basis of the key of the terminal node renewed, and therefore, the device corresponding to the leaf of the child entity not holding the renewal key disables decrypting of EKB produced by the key distribute censer (KDC).

While in the foregoing, the revoking process of the entity in the lowest stage for controlling the device has been described, processing for revoking the entity control entity in the middle sage of the tree by the host entity is also enabled by the process similar to that described above. By revoking the entity control entity in the middle stage, a plurality of entities and devices belonging to the lower level of the entity control entity revoked can be revoked collectively.

As described, by the execution of revoking in an entity unit, revoking process which is simple as compared with the revoking process for executing it in a device unit one by one becomes enabled.

[Capability control of entity]

Next, a description will be made of the processing constitution in which in the key distribution tree constitution in an entity unit, capability granted by each entity is controlled to carry out content distribution according to the capability. The capability termed herein is, for example, defined information of the data processing ability of a device whether decrypting of specific compressed voice data is enabled, whether specific voice reproducing system is granted, or specific image processing program can be processed, whether a device is a device capable of processing what

content or program.

FIG. 41shows an example of the entity constitution which defines the capability. This is the constitution in which a root node is positioned at the uppermost top of the key distribution tree, a plurality of entities are connected to the lower layer, and each node has a 2-branch. Here, for example, an entity 4001 is defined as an entity having capability to grant either voice reproducing systems A, B or C. Concretely, for example, where music data compressed by voice compressed program A, B or C system are distributed, processing for extending the device belonging to the entity constituted below the entity 4001 is enabled.

Similarly, entity 4002, entity 4003, entity 4004, and entity 4005 are respectively defined as entities having capability capable of processing voice reproducing system B or C, voice reproducing system A or B, voice reproducing system B, and voice reproducing system C, respectively.

On the other hand, an entity 4021 is defined as an entity to grant image reproducing systems p, q, r, and an entity 4022 and an entity 4023 are respectively defined as entities having capability to enable image reproducing of a system p.

The capability information of the entities as described is controlled in the key distribute center (KDC). For example, where a content provider desires to distribute music data compressed by a specific compression program to various devices, an enabling key block (EKB) decodable with respect to only the device which can reproduce the specific compression program can be produced on the basis of

capability information of each entity. The content provider for distributing contents distributes a content key encrypted by the enabling key block (EKB) produced on the basis of the capability information and distributes compressed voice data encrypted by the content key to the devices. By the provision of this constitution, it is possible to provide accurately a specific processing program to only the device capable of processing data.

While in FIG. 41, the constitution in which capability information is defined in connection with all the entities is shown, it is noted that it is not always necessary to define the capability information with respect to all the entities as in the constitution of FIG. 41, but the constitution may be employed in which for example, as shown in FIG. 42, capability is defined with respect to only the entity in the lowest stage to which the device belongs, capability of the device belonging to the entity in the lowest stage is controlled in the key distribute center (KDC), and the enabling key block (EKB) that can be decrypted merely for the device capable of providing a process desired by a content provider is produced on the basis of capability information defined in the entity in the lowest stage. FIG. 42 shows the constitution in which capability in entity 4101 = 4105 for which the device is defined, is defined in the terminal node, and capability with respect to these entities is controlled in the key distribute center (KDC). For example, to the entity 4101 belong devices capable of processing a system B with respect to voice reproducing and a system r with respect to image reproducing, respectively. To the entity 4102

belong devices capable of processing a system A with respect to voice reproducing and a system q with respect to image reproducing, respectively.

FIGS. 43A and 43B show an example of the constitution of a capability control table controlled in the key distribute center (KDC). The capability control table has the data constitution as shown in FIG. 43A. That is, propriety with respect to various data processes is set to [1] or [0] such that there are an entity ID as a discriminator for discriminating entities and a capability list indicative of capability defined in the entities, and in the capability list, as shown in FIG. 43B, for example, if a voice data reproducing processing system (A) is can be processed, [1] appears, if not, [0] appears, and if a voice data reproducing processing system (B) can be processed, [1] appears, if not, [0] appears. The method of setting capability is not limited to such a form as described, but other constitutions may be employed if capability with respect to the control device of entities can be discriminated.

In the capability control table, where sub-EKB if each entity of sub-EKB is stored in a separate data base, discrimination information of sub-EKB is stored, and sub-root node discrimination data of each entity is stored.

In the key distribute center (KDC), for example, only the device capable of reproducing a specific content produces a decodable enabling key block (EKB) on the basis of the capability control table. The processing for producing the enabling key block on the basis of capability information will be described with reference to FIG. 44.

First, in Step S4301, the key distribute center (KDC) selects an entity having the designated capability from the capability control table. Concretely, for example, where a content provider desires to distribute reproducible data on the basis of the voice data reproducing processing system A is set to [1] is selected from the capability list of FIG. 43A. an entity, for example, in which item of the voice data reproducing (system A) is set to [1], is selected from the capability list of FIG. 43A.

Next, in Step S4302, a list of selected entity ID constituted by the selected entities is produced. Next, in Step S4303, a pass (a pass of key distribution constitution) necessary for a tree constituted by selected entity ID is selected. In Step 4304, whether or not all pass selections contained in the list of selected entity ID are completed is judged to produce a pass in Step S4303 till completion. This means the process for sequentially selecting the respective passes where a plurality of entities are selected.

When all pass selections contained in the selected entity ID are completed, the procedure proceeds to Step S4305 to constitute a key distribution tree structure constituted merely by the selected entities.

Next, in Step S4306, renewing of node keys of the tree structure produced in Step S4305 is carried out to produce renewal nod keys. Further, sub-EKB of the selected entities constituting the tree is taken out of the capability control table, and the enabling key block (EKB) that can be decrypted merely in the device of the selected entities is produced on the basis of the sub-EKB and the renewal node key

produced in Step S4306. The enabling key block (EKB) thus produced is utilized merely in the device having specific capability, that is, being a decodable enabling key block (EKB. For example, a content key is encrypted by the enabling key block (EKB), and a content compressed on the basis of a specific program in the content key is distributed to the device whereby the content is utilized only in the specific decodable device selected by the key distribute center (KDC).

As described above, in the key distribute center (KDC), for example, only the device capable of reproducing the specific content produces the decodable enabling key block (EKB) on the basis of the capability control table. Accordingly, where a new entity is registered, it is necessary to previously obtain capability of a newly registered entity. The processing of notifying capability with the entity new registration will be described with reference to FIG. 45.

FIG. 45 is a view showing a capability notice processing sequence where the new entity is participated in the key distribution tree constitution.

The new (child) entity (N-En) added newly to the tree constitution executes a new registration request with respect to the hose entity (P-En). Each entity holds a public key in accordance with the public key encryption system, and the new entity sends own public key to the host entity (P-En) when the registration request takes place.

The host entity (P-En) which received the registration request transfers the public key of the new (child) entity (N-En) received to the certificate authority (CA),

and receives a public key of the new (child) entity (N-En) to which a signature of CA is added. These procedures are carried out as the procedure of mutual authentication between the host entity (P-En) and the new (child) entity (N-En).

When the authentication of the new registration request entity is finished by these processes, the host entity (P-Ne) grants the registration of the new (child) entity (N-En) to transmit a node key of the new (child) entity (N-En) to the new (child) entity (N-En). This node key is one node key of the terminal node of the host entity (P-En) and corresponds to a top node of the new (child) entity (N-En), that is, a sub-root key.

When transmission of this node key is finished, the new (child) entity (N-En) constructs the tree constitution of the new (child) entity (N-En) , sets the sub-root key of the top node received to the top of the constructed tree, sets keys of each node and leaf, and produces the enabling key block (sub-EKB) in the entity. On the other hand, the host entity (P-En) also produces the sub-EKB in the host entity (P-En) to which is added a terminal node to be effective by the addition of the new (child) entity (N-En).

When the new (child) entity (N-En) produces sub-EKB constituted by a node key and a leaf key in the new (child) entity (N-En), the new (child) entity (N-En) transmits it to the host entity (P-En), and further notifies capability information with in connection with the device controlled by own entity to the host entity.

The host entity (P-En) which received sub-EKB and capability information

from the new (child) entity (N-En) transmits sub-EKB and capability information received, and renewed sub-EKB of the host entity (P-En) to the key distribute center (KDC).

The key distribute center (KDC) registers the sub-EKB and capability information of entity received in the capability control table described with reference to FIGS. 43A and 43B, and renews the capability control table. The key distribute center (KDC) is possible to produce various forms of EKB, that is, EKB that can be decrypted merely by the entity having specific capability or devices.

The present invention has been described in detail with reference to the specific embodiments. However, it is obvious that those skilled in art may amend or replace the embodiments within the scope not departing from the subject matter of the present invention. That is, the present invention has been disclosed in the form of illustration and should not be interpreted imitatively. For judging the subject matter of the present invention, reference should be made to the claims described herein after.

Industrial Applicability

As described above, according to the information processing system and method according to the present invention, in the production of the enabling key block (EKB) that can be applied as the encrypting processing key block such as a content key, an authentication key, a content check value producing key, a program

data or the like, the hierarchical key distribution tree is reconstructed according to the distribution device, and the enabling key block (EKB) is produced on the basis of the node and leaf contained in the reconstructed simplified tree. Therefore, a considerable reduction of data quantity of the enabling key block (EKB) is realized.

Further, according to the information processing system and method according to the present invention, the enabling key block (EKB) on the basis of the simplified reconstructed hierarchical tree is constituted, and data for judging the propriety of encrypted key data is contained in a tag as a position discriminator of encrypted key data in EKB. Therefore, a considerable reduction in data quantity of EKB is realized, and extraction of encrypted key data using a tag in the device which received EKB is facilitated to make EKB decrypting process in the device effective.